IT'S NOT POLITE TO POINT: REFERRING EXPRESSIONS FOR VISUAL SCENES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Amir Sadovnik

August 2014

IT'S NOT POLITE TO POINT:

REFERRING EXPRESSIONS FOR VISUAL SCENES

Amir Sadovnik, Ph.D.

Cornell University 2014

Recent years have seen a growing interest in generating natural language sentences from images. Most works aim at generating sentences which give a general description of an image. However, this is obviously not the only type of description which can be generated. As has been shown previously in the natural language generation (NLG) community, a full NLG system would require other types of expressions to be generated as well. In this work we present an algorithm to generate referring expressions for natural images. Referring expressions have been investigated extensively in the NLG community since they are an important building block of an NLG system. This thesis presents a novel approach for generating this type of expression taking into account the additional complexities that arise from using natural images. More specifically we focus on issues of saliency, uncertainty due to imperfect attribute classifiers and object detectors, location and relative attributes . By collecting human evaluations we show that our referring expressions are useful in referring viewers to a specific image or a specific object within an image.

This document is dedicated to my wife and kids.

I couldn't ask for a more supportive and loving family.

ACKNOWLEDGEMENTS

First I would like to thank my advisor Prof. Tsuhan Chen for his support throughout my time as a PhD student. The guidance he has provided me during our weekly meetings served as a compass for my research, and ensured I was staying on track with my work and focusing my energy in the right direction. In addition, the freedom he provided in choosing my own research topics was invaluable and allowed me to do research in subjects which I was truly interested in and which I could call my own. He also made sure that at all times I was understanding the bigger picture in which my thesis fits in.

I would like to thank Prof. Shimon Edelman from the psychology department for the many conversations we had early on in my PhD studies. Many of the ideas we discussed have ended up shaping important parts of my thesis and others will help shape my future research directions. Having a psychology professor who is interested in both computation and vision allowed me to have an external view on my research ideas which is very important.

I would like to thank Prof. Anthony Reeves and Prof. David Forsyth for sitting on my committee and always providing insightful comments and ideas for improvements for my thesis work. Many of these ideas made way into the current version of the thesis and others will provide directions for future work.

I would like to thank Dr. Andrew Gallagher for collaborating with me closely on much of my work. The fact that his office was always open and his willingness to discuss any idea in depth really assist me during the latter part of my work. The help in both ideas and writing is greatly appreciated.

I would like to thank other collaborators and fellow lab members: Prof. Noah Snavely, Prof. Devi Parikh, Prof. Druhv Batra, Congcong Li, Zhaoyin Zia, Yimeng Zheng, Adarsh Kowdle, Henry Shu, Kuan-chuan Peng, Amandi-

aneze Nwana, Ruogu Fang. The many meetings both in person and through Skype kept helping me improve my research and pushed it to where it is today.

Finally I would like to thank my wife and kids. Their support and love is what truly allowed me to dedicate all these years to this work and for that I am truly grateful.

TABLE OF CONTENTS

Dedication 4
Acknowledgements 5
Table of Contents 7
List of Tables 9
List of Figures 10

1 Introduction **1**

2 Referring Expressions for Scenes **6**
2.1 Introduction 6
2.2 Previous Work 9
2.3 Item Detection 12
2.3.1 Object Collection 12
2.3.2 Relationship Detection 13
2.3.3 Color Detection 13
2.4 Item Ranking 14
2.4.1 Item Probability 14
2.4.2 Salience 15
2.4.3 Combining the scores 16
2.5 Constructing the Sentences 17
2.6 Experiment Design 18
2.7 Results & Discussion 21
2.7.1 Discriminating Description 22
2.7.2 Parameter Evalutation 25
2.8 Future Work 28

3 Referring Expressions for Objects **30**
3.1 Introduction 30
3.1.1 Previous Work 34
3.2 Considering Attribute Uncertaintity 37
3.2.1 Attribute detection 37
3.2.2 Guesser's Model 39
3.2.3 Single Attribute 41
3.2.4 Multiple Attributes 43
3.2.5 Guesser-Based Attribute Selection 44
3.3 Considering absolute location 45
3.4 Considering Relative Location 49
3.4.1 Considering Relative attributes 50
3.5 Experiments and Results 52
3.5.1 Computer Baselines 53
3.5.2 Human Describers 56
3.5.3 Absolute Location Results 57

3.5.4 Relative Location Results 59
3.5.5 Relative Attributes Results 63
3.6 Conclusion 65

4 Conclusion **67**
4.1 Future Work 68
4.1.1 General Image Description 68
4.1.2 Adding Additional Attributes 69
4.1.3 Expanding to Other Object Classes 70
4.1.4 Applications 71

A Worked Out Example **73**
A.1 Single Attribute 73
A.2 Multiple Attributes 75

Bibliography **78**

LIST OF TABLES

3.1 Variable definitions 38

3.2 Results of our three different experiments as described in Sec. 3.5. (a) and (b) use the presentation method as shown in Fig. 3.12(b), while (c) uses the presentation method as shown in Fig. 3.12(c). The last row is the sum of the first two, and signifies the total percentage of people who chose the true target/anchor as one of their choices. 60

A.1 The Poisson Binomial PMF of the number of faces with the correct attribute. 74

A.2 Two attributes example 76

A.3 Three Poisson Binomial PMF's, one for each face, over the number of attributes correct for that face. 76

A.4 An example of a table akin to Table A.1 for the meta attribute "Satisfying two attributes". 77

LIST OF FIGURES

1.1 A screen shot of our experiment in Chapter 2. The subject is given a referring expression generated by our algorithm and must select the correct scene. 3
1.2 Most previous work on REG in the NLG community start with a problem as shown on the left, where the attributes for each entity are known. However, since the computer uses imperfect features and an imperfect classifier, it is necessary to deal with probabilities as is shown on the right. 4

2.1 In this chapter we develop a method for creating the most efficient textual description that can discriminate one image from a group of images. For example, if the image with the red border is our target image and the rest are distractors, an efficient description might be: *"The image with the gray oven"*, since the target image is the only one in which a gray oven exists. The white cabinets or the window would not need to be mentioned, because they exist in other images. The plant, which also only exists in the target image, might be harder to find because of its smaller size and thus does not need to be mentioned as well. . . 7
2.2 Our approach to building a discriminating description. Given a target image and a set of distractors, we first build a graph for each of the images with three different types of nodes: (a) objects (b) relationships (c) colors. Then, using the graphs from all the images, we rank the different items in the target image. This ranking is based on two main criteria: discriminability and salience. Finally, depending on the length of description we require, we use the top n items and submit them to a natural language generator to create the final description. 11
2.3 An illustration of why visual saliency should be helpful. When trying to build a description for image (a) the apple is the most discriminative object. However, it is small and might be missed. On the other hand, although a chair exists in both images, it is much more salient in our target image. Therefore, if we choose to describe the chair instead of the apple, we should be able to distinguish the target image. 15
2.4 Four examples of the output of our discriminative description for different target images from different categories (living room (a)+(d), kitchen (b), bathroom (c)). Although the distractors are not shown here, each item described is chosen by being the most discriminative (no saliency). 19

2.5 A screen shot of our experiment. The subject is given 10 images and instructed to select the one described by the algorithm. The subject also has the option of enlarging any of the images. In this specific case, the target image is the one on the bottom right. . . 21
2.6 Results from our lab experiment. The x axis represents the number of items in a description, while the y axis represents the percentage of subjects who succeeded in guessing the correct image when less than x items were given. 22
2.7 Average time to guess correctly. For each description length (up to three) we take only the subjects who have guessed correctly and calculate the average time they spent guessing at that description length. 24
2.8 The results of our three Amazon Mechanical Turk experiments. In each experiment we examined the effect of one of the parameters and set the other two parameters to 0. (a) The effect of α which is the weight given to the size of the object. (b) The effect of β which is the weight given to the low level saliency of the object as described in the model by [17]. (c) The effect of γ which is the weight given to the centrality of the object. 26
2.9 Image examples of how saliency can assist in discriminating between images. The colors represent different values for the different parameters, while the graph shows the improvement in performance for each parameter for that specific image. (a) On the left of the image there is a small basket above the sink. This is very hard to notice. However, the plant next to the cabinet in the front-right of the image is much easier to see and therefore provides a 25% increase in guessing. (b) There is a cup in the middle of the image. However, since it is clear it has very few edges. Although the outlet is small it has a much higher saliency score and thus provides a 30% increase. (c) Although both the carpet and the curtain only existed in this image out of all the distractors, the curtain is centered, so it provided a 12% increase. (d) Although using the size parameter helps in choosing the carpet over the basket, if it is too high then too much weight is given to the size and it selects a non-discriminative item. 27

3.1 In this chapter we introduce an efficient method for choosing a small set of noisy attributes needed to create a description which will refer to only one person in the image. For example, when the target person is person *(b)*, our algorithm produces the description: "Please pick a person whose forehead is fully visible and has eyeglasses" 31

3.2 An overview of our algorithm. (a) Given an image of a group of people (b) detect all faces and select a random target. (c) For each face run a set of attribute classifiers. (d) Find a small set of attributes which refers to the target face with confidence c (e) Construct a sentence and present to a guesser. 34
3.3 An illustration calculating the probability of guessing correctly using one attribute ("The person is smiling") for an image with three people. The true identity of the target person (marked with a red rectangle) is known to the algorithm as well as the attribute confidence for each face. Each face is actually smiling or not (the true state is unknown to the algorithm), represented with the blind over each mouth. To find the probability of the guesser's success, each of the eight possible configurations of smiling faces is considered. We introduce a polynomial-time algorithm for computing this probability. 40
3.4 An example of transforming the table of p_{ki} into the 4 PMF's of y_i (one per column). In Eq. 3.8, j iterates through the different rows and normalizes accordingly. 43
3.5 Fitting a logistic function to the location based attributes. 46
3.6 An example of face rows detected in the image. 47
3.7 Two images with row based descriptions vs. attribute based descriptions. The examples clearly show that for some faces row based descriptions would be useful while for others they would not. (Description in green is assessed to be a better referring expression) 49
3.8 A comparison of a binary SVM and Rank-SVM. In our work we normalize the difference obtained by Rank-SVM to a probability that one person has more of an attribute than another face. 51
3.9 Our results from the computer baseline experiment (Sec. 3.5.1). (a) Guessing accuracies for the five methods introduced in Sec. 3.5.1. 1. confident 2. top_used 3. Full_greedy 4. GBM (b) Accuracy results of GBM as we increase the minimum threshold, by looking at descriptions whose confidence level as calculated in Eq. 3.8 are higher than it. (c) The percentage of descriptions (methods 1-4) an attribute was used in for a select set of attributes. The attributes are: (1) Gender (2) White (3) Black hair (4) Eyeglasses (5) Smiling (6) Chubby (7) Fully visible forehead (8) Eyes open (9) Teeth not visible (10) Beard 54

3.10 Examples of our GBM algorithm along with the calculated confidence and the actual accuracy received from AMT. The left two are examples where our algorithm correctly estimates the confidence (approximately). The right two examples are failure cases: A misclassified target attribute (no hat on target) and a misclassified distractor attribute (additional bearded person in the image). 55
3.11 Examples of the different descriptions created using Full_greedy vs GBM, and the accuracy achieved in our collected results. The GBM method realized that mentioning *gray hair* after *is senior* is unnecessary and managed to choose a more important 55
3.12 Examples of 3 different ways we presented our descriptions. (a) Text: an exclusively textual description as in [31]. (b) Graphical: Our graphical representation (c) Two-Step: Our two step presentation. The second part is only shown after the first part was completed. 59
3.13 Examples of the different descriptions created using GBM vs GBM_neighbors*, and the accuracy achieved in our collected results. In these examples it is clear to see that since it is hard to differentiate the target person from the distractors, using a neighbor anchor face clearly simplifies the task. 60
3.14 Guessing results using our original algorithm without superlatives vs. our original algorithm with the superlative attributes. . 66
3.15 Examples adding superlatives as attributes to our original framework. For each image the left column shows the attributes selected by the algorithm when no superlative attributes where available, while the right column shows the ones in which superlative attributes were available. The green column shows two examples in which adding superlatives was helpful, while the red row shows two examples where adding superlatives produces lower quality referring expressions. 66

CHAPTER 1

INTRODUCTION

Human-Machine collaboration has been the focus of many recent works in both the robotics and computer vision field. The general idea is to move away from the strict paradigm of the past in which machines simply understand and perform instructions given by humans. Instead, the we wish to allow the human and the machine to communicate on a given task in order to perform it optimally.

There are many reasons why a machine would find it beneficial to communicate with humans. For example, imagine a robot housekeeper scenario, in which a robot is tasked with setting up a table for dinner. Although his main tasks would require it to understand your instructions and set the table as you wish, there are many reasons the robot would need to communicate with you in order to perform his main task better. It might want to inform you regarding difficulties it is encountering, it might need to ask a clarification question regarding your instructions, or in cases where it does not understand a term you are using it might need to ask a question in order to learn a new word. All these would require the machine to be able to generate natural language sentences.

Most of the traditional research in computer vision has focused on providing tools for machines to be able to perform different tasks without examining this two way communication. Tools such as classification and detection allow users to search for relevant images. Tools for 3d scene understanding allow robots to navigate complex 3d scenes. However, in order for a machine to be able to generate language it requires other skills as well. The machine would need to perform the basic tasks for understanding, but additional natural lan-

guage building blocks such as lexicalization, ordering and aggregation would be necessary for it to convey the correct message.

In this work we focus on one of these important building blocks: Referring Expression Generation (REG). A referring expression is an expression which enables the listener to identify a given object within some context. These expression are an important part of any communication especially ones which have to do with preforming tasks. For example, if we return to our robot housekeeper example, imagine that you asked the robot to set the green porcelain dish-set for dinner. The robot might respond by saying (the expressions in bold face are referring expressions):

- Would you like me to set up **dining room wooden table** or **the kitchen glass table**?
- Is the porcelain set **the one on the top shelf on the right**?
- It might be preferable to use **the red plastic plates** for the kids.

As can be seen, all those interactions would require an REG module. However, since REG is a well studied topic within the natural language generation (NLG) community, why is there a need to revisit it within a visual context? The answer is that there are many additional factors which arise when dealing with real world visual scenes that have not been thoroughly examined previously in the NLG community. Although topics such as uncertainty, salience, relative and absolute location have been addressed at times in previous works, none has dealt with addressing them specifically for visual scenes.

Our work attempts to fill in this gap. We do so by addressing the two main problems. First, in chapter 2 we address the generation of referring expressions

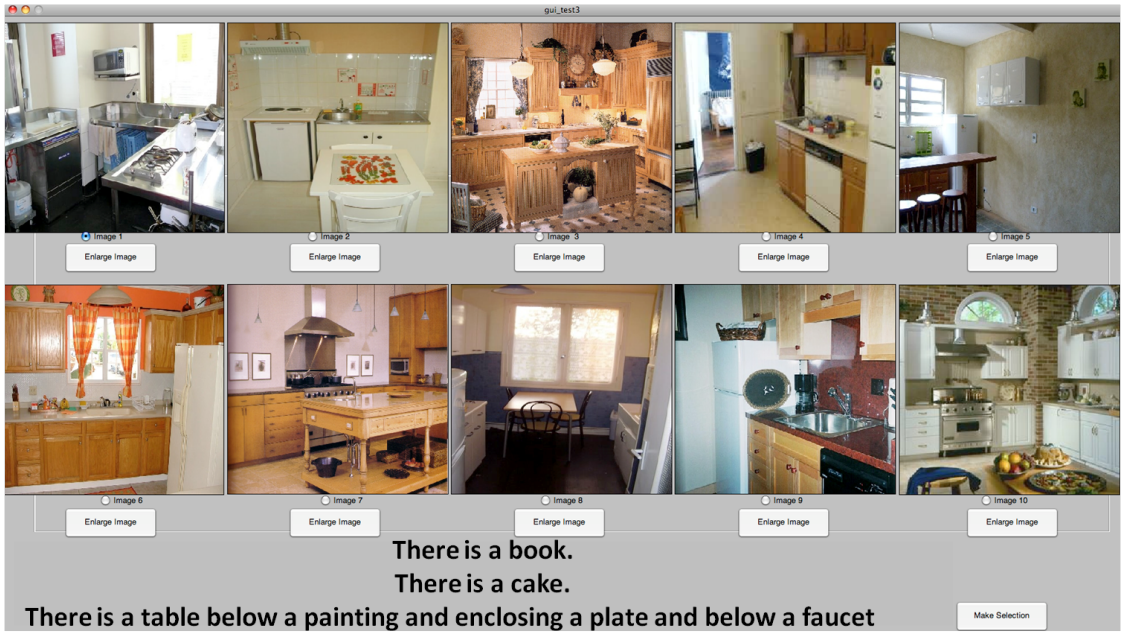

Figure 1.1: A screen shot of our experiment in Chapter 2. The subject is given a referring expression generated by our algorithm and must select the correct scene.

for visual scenes. The task is to describe a scene in terms of the objects it consists of in the context of other scenes as shown in Fig. 1.1 . In this chapter we focus mainly on different measures of object saliency in a scene such as centrality, size and low level saliency, and examine how incorporating these measures into our REG algorithm effect the results. This work was originally in published in [33].

In chapter 3 we examine referring expressions for objects within scenes. More specifically we attempt to generate referring expressions for people in images with groups of people. Working with people gives us the advantage of having a wide attribute vocabulary to select from, and therefore create more complex descriptions which allow us to investigate this topic more thoroughly. In this chapter we mostly focus on the problem of uncertain attributes. That is,

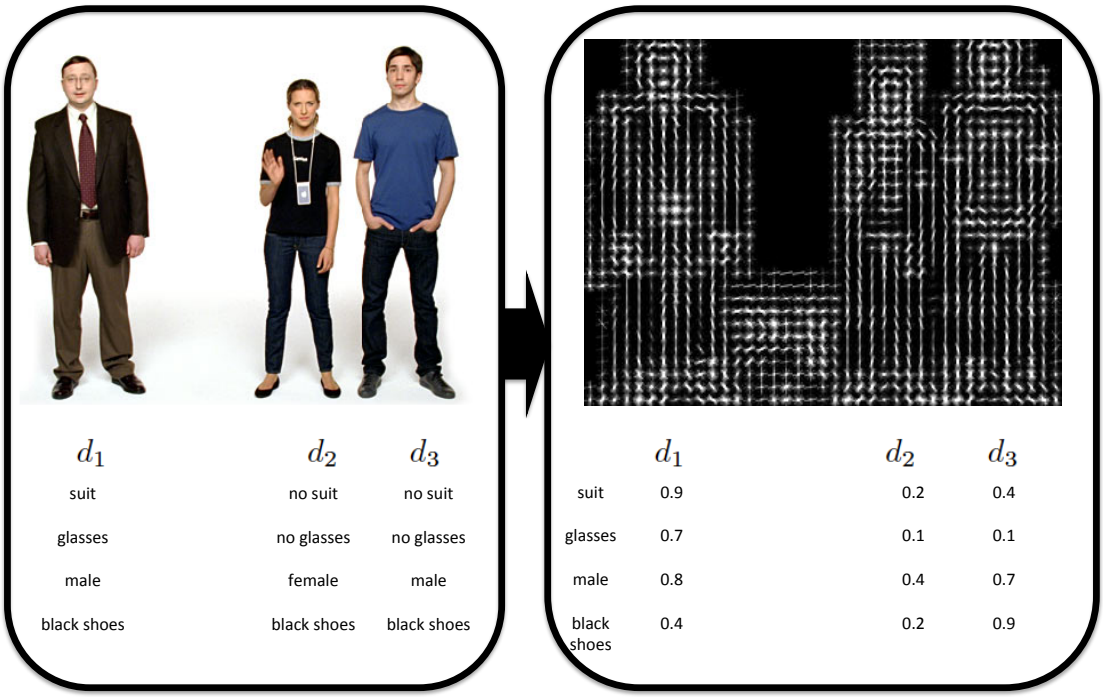

Figure 1.2: Most previous work on REG in the NLG community start with a problem as shown on the left, where the attributes for each entity are known. However, since the computer uses imperfect features and an imperfect classifier, it is necessary to deal with probabilities as is shown on the right.

since our attribute classifiers are not perfect, the computer has to generate the referring expressions based on attribute probabilities versus attribute lists (as shown in Fig. 1.2) . This introduces additional complexities which we manage to solve and show that we are able to generate robust referring expressions. This work was originally published in [31]

In this chapter we introduce additional extensions to the basic framework. First we examine both absolute and relative location. In a visual scene, in addition to appearance based attributes we also have the location of our target object, and would expect this information to help in generating referring expressions.

We examine two ways to use location. First, we incorporate the absolute location of the object in the image by adding location based attributes to our original framework. Then we examine the notion of relative location by looking at the target object's neighbors, and evaluating if it would be preferable to describe them and the spatial relation of the target to them versus simply describing the target itself. This work was originally published in [32]

An additional extension we examine is the use of relative attributes. In our basic framework attributes are only used in their binary form. However, attributes can be used in additional ways as well, such as in relative form. That is, instead of simply stating if a person is smiling or not, we can say that they are smiling less/more than someone else. We introduce this into our original framework by adding additional attributes in the form of superlatives such as "the most smiling".

Finally, in chapter 4 we discuss future research directions and applications to the work presented.

CHAPTER 2

REFERRING EXPRESSIONS FOR SCENES

2.1 Introduction

Scene understanding is one of the ultimate goals of computer vision. However, coming up with methods to attain this goal is still a very hard problem. Most of the computer vision field is currently focused on trying to extract the information needed for scene understanding such as object detection, scene recognition, and 3D modeling. However, merely listing the output of these algorithms would not amount to a true understanding of the scene. To show a higher level of understanding, one may try to rank these outputs so as to describe things in a correct order and omit those that are of no import.

A visual scene may contain a large number of objects. All these objects stand in certain spatial relationships with one another, and to each one we might be able to ascribe many attributes. However, when asked to describe an image, a human viewer will obviously not choose to name all such objects one by one. Indeed, there is likely to be a great deal of information that a human will deem unnecessary to mention at all. This is partly because a genuine understanding of a scene makes certain items very important, while rendering others insignificant. In addition, different tasks might require different descriptions of the scene. A general description of a scene might be different from a description aimed at singling out one image from a group of images. In this work, we focus on the latter task. As far as we know, this is the first attempt to construct such discriminative description for general scenes.

Figure 2.1: In this chapter we develop a method for creating the most efficient textual description that can discriminate one image from a group of images. For example, if the image with the red border is our target image and the rest are distractors, an efficient description might be: *"The image with the gray oven"*, since the target image is the only one in which a gray oven exists. The white cabinets or the window would not need to be mentioned, because they exist in other images. The plant, which also only exists in the target image, might be harder to find because of its smaller size and thus does not need to be mentioned as well.

Consider for example Fig. 2.1, where the task is to distinguish the image framed in red from the others. If we merely create a "laundry list" of all the objects in the image with their colors and relationships, we might end up with a type of description that starts as follows (we omit the ending of the description because of its length):

"There is window above a gray sink. The sink is above a white cabinet which is next to a white dishwasher. There is a gray oven below a gray stovetop next to a white drawer. There are brown chairs next to a table. . . ."

However, if our task is simply to discriminate our target from the other images, we should be able to use a description as simple as:

"There is a gray oven"

This description is much more efficient for this specific task in that it con-

veys the same amount of information in many fewer words. In this work, we investigate the possibility of creating such efficient descriptions automatically.

Although this is a specific task, it is useful for other tasks as well. For example, when describing an image, it is known that people tend to mention the unexpected. Therefore, this type of task in which we specifically search for what is unusual about an image as opposed to other ones that are similar to it will need to be incorporated in any system whose goal is to create natural sounding descriptions.

By choosing this specific task, we are also able to measure the effectiveness of our description in a more quantitative manner. This is in contrast to previous works ([8],[35]) in which results are mostly assessed qualitatively. We show that by ranking the candidate items according to our new metric, we are able to create shorter and more efficient descriptions. In addition, we show how the different factors we use to rank these items contribute to the performance.

Although we construct a textual description, in this work we do not focus on the sentence structure. We instead focus on using the visual data to rank the different items in the image, and so create very simple sentences based on specific rules. Although creating an appropriate grammar is also an important part of the challenge of image description, we believe that it is mostly independent of the visual data on which we concentrate. That is, given a set of items and relationships that need to be described, the description and the image are independent. Therefore, by providing the item information to a more complex natural language generation algorithm, a more realistic description can be created.

2.2 Previous Work

In recent years, there have been a few attempts to develop automatic methods for generating textual descriptions of images. For example, Farhadi et al. [8] try to use existing descriptions from the web and match them to new images. Although this method, which constitutes one of the first attempts at scene description, can associate natural sounding sentences with images, it is limited in that it can only select a description from a given database of sentences. Therefore, given a new image, the probability of finding a sentence that describes it very closely is relatively low.

Yao et al. [39] also try to create a textual image description for a scene. They use a hierarchical parsing of the image to generate the description. They try to learn a complex model in which knowledge base information from the web is used to parse the image and to create sentences. While this allows them to generate more natural-sounding sentences, they do not attempt to filter the information detected in the image, and simply end up describing everything.

Berg et al. [21] uses a conditinal random field to detect different objects/relationships/attributes in images. This CRF uses textual descriptions from different online databases to encourage the detection of commonly used objects/relationships/attributes combinations. This means that if a certain relationship was mentioned many times in the database, this relationship will be encouraged in the CRF. Although this approach does take into account the probability of mentioning certain items, it does not do so on a per-image basis. Since all images use the same description database as potentials in the CRF, a certain item may be encouraged regardless of the specific image being described. In

our approach, we attempt to tailor the description to a specific image for a specific task. For example, if in the online database no one ever mentioned a "blue cat", this attribute-object relationship will be discouraged. However, given an image with a "blue cat", this might be exactly what we would want to describe because of its unusualness.

Among other related works, Spain et al. [35] ask people to name objects in photographs, then use this informastion to build a model that tries to predict the importance of objects in novel images. Although this task resembles ours, there are two main differences. First, Spain et al. only consider objects and do not attempt to rank also attributes or relationships. In addition, subjects are asked to list the objects without a specific task in mind. Our work attempts to provide the most efficient description for a specific task.

Farhadi et al. [7] use high-level semantic attributes to describe objects. While most of their work is about object classification and category learning, they do discuss textual description, noting that focusing on unusual attributes results in descriptions similar to those generated by humans. They do not, however, try to combine objects and atttibutes into a scene description. We use the idea of highlighting unusual attributes in our description.

In natural language generation, there has been much work on referring expressions (for an extensive and recent survey see [20]). These are sentences that can refer to one and only one item among a set of items. This work is very closely related to ours, but there are some major differences, which stem from our use of visual data, instead of just a list of properties. For example, imagine trying to refer to the first scene out of the following two:

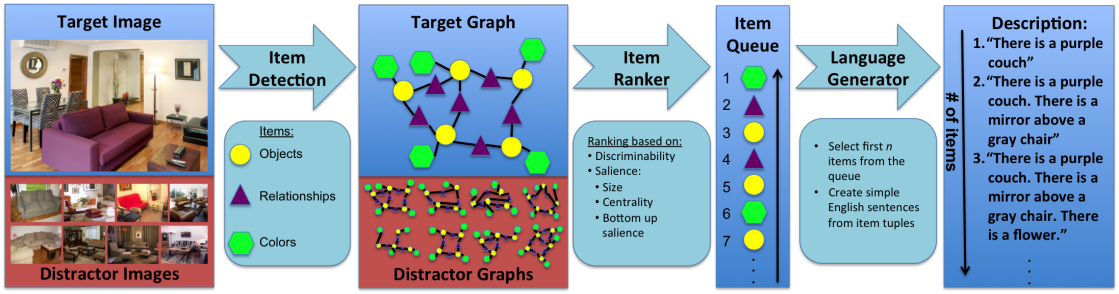

Figure 2.2: Our approach to building a discriminating description. Given a target image and a set of distractors, we first build a graph for each of the images with three different types of nodes: (a) objects (b) relationships (c) colors. Then, using the graphs from all the images, we rank the different items in the target image. This ranking is based on two main criteria: discriminability and salience. Finally, depending on the length of description we require, we use the top n items and submit them to a natural language generator to create the final description.

- chair, table, apple, melon, strawberries, blueberries.
- chair, table, melon, strawberries, blueberries.

The obvious choice for a referring expression generator would be to describe the apple. However, in Fig. 2.3 we show that this is not the best description when using visual data.

An overview of our method is shown in Fig. 2.2. This method allows us to create an efficient tailored description for a specific set of target/distractor images. In contrast to previous work, our description is goal-oriented and includes a quantitative estimate of its quality.

2.3 Item Detection

Similarly to previous work ([21],[24]), we focus on three main types of visual information that can be used to describe a scene:

1. The objects in the scene $O = \{o_1, o_2, ..., o_n\}$
2. The relationships between the objects $R = \{r_{12}, r_{13}, ..., r_{nm}\}$
3. The colors of each object $C = \{c_1, c_2, ..., c_n\}$

We refer to the unified set of O, R, C as items. In this section, we describe how we collect these items; section 2.4 states our approach to ranking these items.

2.3.1 Object Collection

The main building blocks for our description are the objects that exist in the image and their categories. We use labeled data for localizing and recognizing objects. Since our main focus is not on recognition but on the description task, we would like to be able to have as many objects as possible in an image, coming from a wide range of object categories. We use three different categories from the indoor LabelMe dataset: kitchens, bathrooms, and living rooms [30]. After cleaning up the LabelMe data, we obtain a dataset with over 150 different types of objects, and an average of 20 objects per image. Although labeled images are expensive, this gives us images with a much richer variety of categories than those used before for image description tasks, allowing us to assess the quality of our algorithm under much more interesting conditions.

2.3.2 Relationship Detection

We focus on three types of relationships between objects: "above", "overlapping", and "next-to". To detect these, we simply calculate the relative position (Δx, Δy) and overlap (O) between all pairs of objects that are less than a certain number of pixels away from each other. We then use the following criteria to define the relationship:

1. A "overlaps" C if $\frac{O_{AB}}{BB_A} > 0.8$ where O_{AC} is the overlap area and BB_A is the bounding box area of A.
2. A is "above" C if $-0.375\pi < \tan(\frac{\Delta y}{\Delta x}) < 0.375\pi$
3. A is "next-to" C for all other objects whose distance is less than the threshold.

2.3.3 Color Detection

Among various possible attributes of an object, we choose to detect color, since it offers fairly reliable results. Our color classifier distinguishes among 11 different colors, using the database of [37]. As features we use a normalized binned histogram in HSV space [41]. We then use an SVM with an RBF kernel [3]. When presented with a new set of images, we use the mask of each object to extract the feature histogram. Then after running the classifier, we get a set of 11 probability values (one for each color), which signify the likelihood of the object to have that specific color.

2.4 Item Ranking

Our description model resembles the incremental algorithm of [5] for referring expression generation. The basic idea is that when people use a referring expression to describe an object, they have a certain preference in mentioning certain items over others. This preference order can be viewed as a queue in which all the items are waiting. By going through these items one by one, the speaker iteratively selects the ones that are discriminative enough under some criteria (for example, those that can eliminate more than n objects). Our goal is to construct the item queue from visual data. We do this by calculating a score for each detected item, and then sorting them in decreasing order.

2.4.1 Item Probability

The first property of the item we examine is its discriminability: given a set of images I, including our target image, we calculate the probability of the item being in this set. This obviously captures the discriminability of the item, since the lower the probability, the more images would be eliminated by including it. More specifically, we calculate the following probabilites:

$$p(cat_i|I) = \frac{|I_{o_i}|}{|I|}$$
$$p(rel_{r_{ij}}|I) = \frac{|I_{r_{ij}}|}{|I|}$$
$$p(col_{c_i}|I) = \frac{|o_{i_c}|}{|o_i|}$$

Where I_{o_i} is the set of images with an object from category i, $I_{r_{ij}}$ is the set of images with relationship r between objects of type i and j, o_{i_c} is the set of objects of type i with color c and o_i is the set of objects of type i.

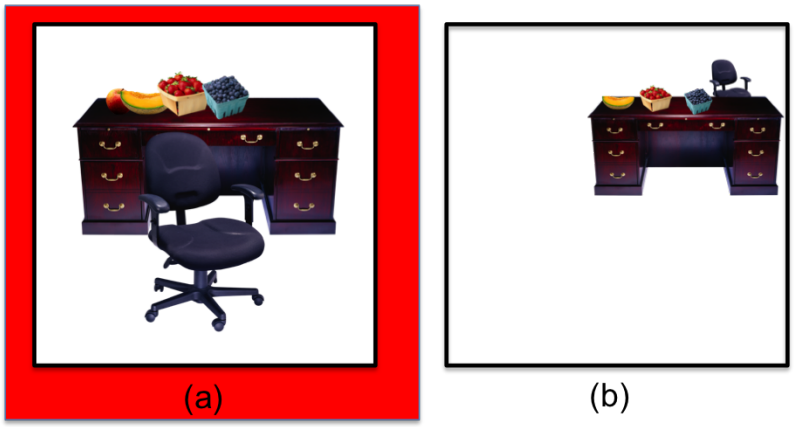

Figure 2.3: An illustration of why visual saliency should be helpful. When trying to build a description for image (a) the apple is the most discriminative object. However, it is small and might be missed. On the other hand, although a chair exists in both images, it is much more salient in our target image. Therefore, if we choose to describe the chair instead of the apple, we should be able to distinguish the target image.

2.4.2 Salience

We note that simply choosing the most discriminative item would not necessarily lead to the best discriminative description. This is because not all visual data are equal. There are many different properties of an item that might make it more or less useful in a description (cf. Fig. 2.3). We therefore use the following three measures for saliency inspired by Spain et al. [35]:

1. Size of the item. We normalize the size of each object by dividing by the size of the image.
2. Low level saliency of the object. A saliency map based on the work of Itti and Koch [17] implemented by [15].
3. Centrality of the item. The distance from the center normalized by the size of the image.

We calculate these values for each of the items. Since the relationship item involves two objects, we use the mean value of the two as the saliency feature for it. Taking this average can prove to be very useful under certain conditions. For example, in Fig. 2.3, the apple is the most discriminative item to describe image (a), since it does not appear in image (b). However, since it is very small, it might be missed. Since the apple is discriminative, the relationship "apple above table" is as discriminative, but has a much larger size score, because the size score is taken from the mean of the apple and the table. Therefore, this relationship might be ranked highest, and the description *"there is an apple above the table"* will be given. This will allow the listener to find the target image much quicker and perhaps avoid missing the apple all together.

2.4.3 Combining the scores

We formulate a score for each item based on its discriminativity and its salience. This score represents the importance of the specific item in the target image as related to the set of images I:

$$Score(IT_i) = (1 - p(IT_i|I)) + \alpha S(O_i) + \beta L(O_i) + \gamma C(O_i) \tag{2.1}$$

Where IT_i is an item, O_i is the object(s) that exist(s) in the item, $p(IT_i|I)$ is one of the probabilities as described in Sec. 2.4.1, and S, L, C are the size, low-level saliency, and centrality respectively. The parameters α, β, and γ are the weights given to the different saliency measures; these need to be adjusted to an optimal value. Too low a value may result in very non-salient items being chosen, which may be discriminative, yet easy to miss. At the same time, too high a value may cause the algorithm to choose items that are salient but not very discriminative.

Although the users would be able to find those items quickly, they may exist in multiple images, and therefore not be of much help. We examine the effect of changing the parameters in Sec 2.7.2.

Our color classifier can produce erroneous results which can cause the user to make mistakes.Therefore, we use the probability score $P(c)$ that is given by the SVM to minimize these types of errors. We multiply $1 - P(c)$ by a fourth parameter δ, and subtract that from the score of the color items in equation 2.1. Colors for which the classifier gave a low probability (low confidence) will therefore get a low score and thus not be mentioned.

Once we have calculated all the scores, we rank the items based on the score in descending order. We then form a description of length n by choosing the n items with the top score. These items can be thought of as a set of n-tuples:

1. $\langle object \rangle$ a single for the object item
2. $\langle object, color \rangle$a double for the color item
3. $\langle object_1, relationship, object_2 \rangle$ a triple for the relationship item.

These n-tuples are then sent to a language generation algorithm in order to construct more natural english sentences.

2.5 Constructing the Sentences

Although we do not focus in this chapter on the task of constructing perfect English sentences from the items we choose, we still need to perform some simple

operations in order to make the sentences understandable and clear for experimentation. We follow a few very basic rules:

1. The first time an item is introduced, construct the sentence: *"there is a* $\langle ntuple \rangle$*"*
2. If an item has been introduced using the relationship or color item, remove the simple introduction of the item ($\langle object \rangle$) from the queue since it would be redundant if intoduced later.
3. When an object exists in more than one relationship, introduce an *"and"* between them and remove $object_1$ from the second triple.
4. Always place the color before the object it describes (even if the object has already been introduced).

There are a few obvious limitations to this approach, which can result in unnatural sentences. First, there is no notion of numbers in this method, and therefore if there are two objects of the same category it simply gets mentioned twice. In addition, there is no notion of continuity between sentences and therefore the transition between them appears unnatural. However, given all these limitations, the description is clear and concise in such a way that the necessary information is conveyed to our subjects. For an example of descriptions created by our algorithm, see Fig. 2.4.

2.6 Experiment Design

We ran an experiment with human subjects to measure how well the descriptions generated by our method can discriminate among a set of images. The

Figure 2.4: Four examples of the output of our discriminative description for different target images from different categories (living room (a)+(d), kitchen (b), bathroom (c)). Although the distractors are not shown here, each item described is chosen by being the most discriminative (no saliency).

experiment is conducted as follows. The computer first selects a random set of 10 images. Out of these, it then chooses a random target image which it tries to describe to the human subject. After detecting and ranking the items from the target image as discussed in sections 2.3 and 2.4, the algorithm presents the 10 images to the subject along with a description that includes only the top scored item. The subject is then required to select the correct image based on the description. If the subject is correct, the trial is over. However, if the subject selects a wrong image, the algorithm takes the next highest ranked item from the list and offers a new description that includes the top two items. This happens repeatedly until the subject selects the correct image, or there are no more items to describe in the image, or the subject has failed a certain number of times.

To examine the effects of different values for parameters α, β and γ, we needed to conduct a larger scale experiment. To that end, we adapted the experiment to work in Amazon Mechanical Turk, with a slight adjustment. The main difference is that each user only gets one chance at guessing the correct image, given a certain length of description. Whether or not the answer is correct, the next picture is then presented. Since we perform these tests with descriptions of different length, we are able to get a complete set of results from this style of testing.

To make the task more challenging and the choices less trivial, we select the distractor images to be of the same scene category as the target image. We end up using three scene categories from the indoor dataset: kitchen, bathroom and living room [30]. We use these scene categories because they contain many different object categories, as well as many object per scene. Thus, if the target image is a kitchen, all the distractors would be images of different kitchens. For

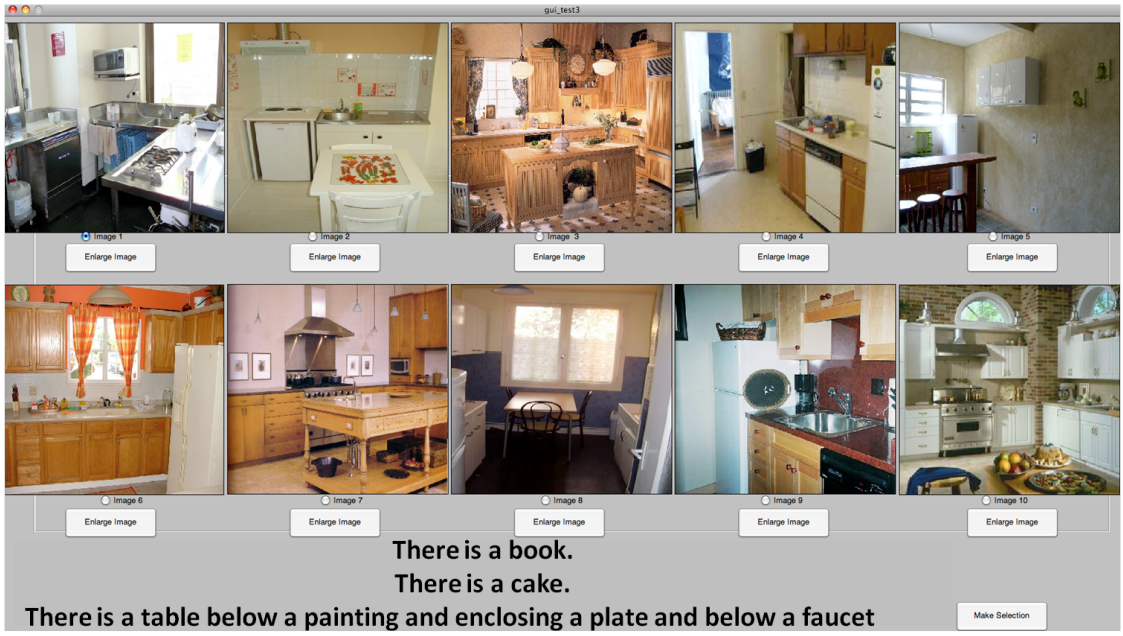

Figure 2.5: A screen shot of our experiment. The subject is given 10 images and instructed to select the one described by the algorithm. The subject also has the option of enlarging any of the images. In this specific case, the target image is the one on the bottom right.

a screenshot of our experiment, see Fig. 2.5.

2.7 Results & Discussion

We divide the description of our results into two sections. We first present the results of the experiments run in the lab. These experiments were used to verify that in our set-up, choosing the most discriminating items (regardless of the other parameters) for the description would allow people to choose the correct image given its length of description with higher percentage. We compare our results to a random selection, in which all the items in the image are ordered

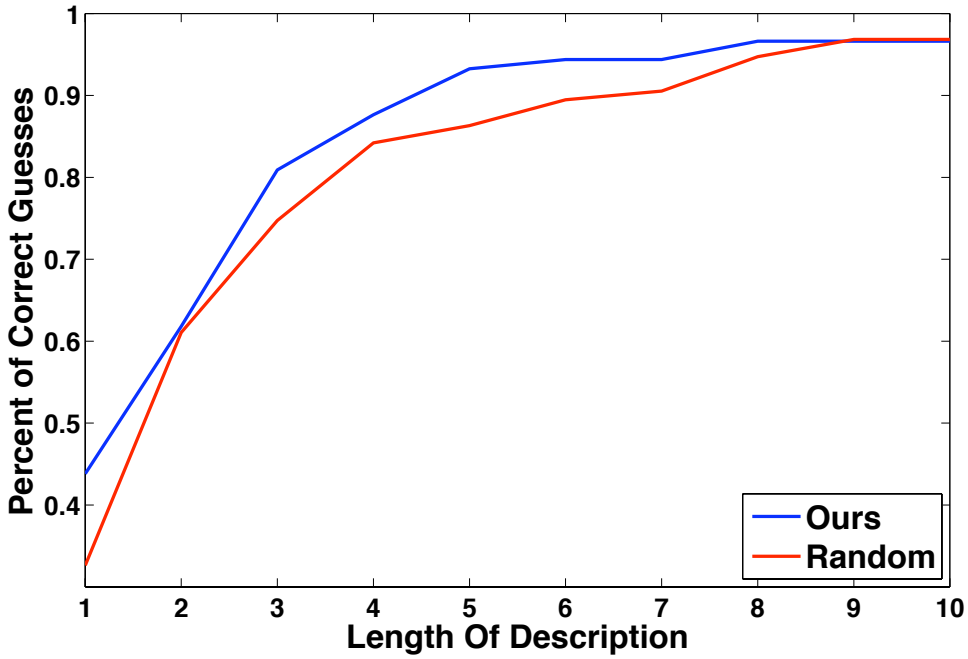

Figure 2.6: Results from our lab experiment. The x axis represents the number of items in a description, while the y axis represents the percentage of subjects who succeeded in guessing the correct image when less than x items were given.

randomly in the queue and then chosen one by one.

2.7.1 Discriminating Description

The lab experiment was performed on 18 subjects, using the kitchen category. Each subject had 15 trials, where each trial is a set of one target image and nine distractors. On average, the subjects needed 2.5 guesses per trial. That is, since every time they guessed wrong they received a longer description for the same trial, they usually needed more than one guess.

The results, presented in Fig. 2.6, show that the discriminative selection yields better performance than random selection. For example, only 32.6% of subjects managed to guess the correct image given a random description with only one item, while 43.8% managed to guess correctly with our discriminative approach ($p = 0.059$). Although the curves do get closer and the difference less significant for certain numbers of items, the discriminative description always results in better performance.

There are a few reasons why the performance in the random description and the discriminative description conditions are relatively close. First, although the images we selected as distractors are all from the same category, they are usually different enough, such that even after a few items it is relatively easy to find the correct image. This is even more pronounced after the subject has already made an incorrect guess, since at that point he or she had already eliminated one of the images. For example, after a wrong first guess the subject's chances increase from 1:10 to 1:9.

In addition there is much noise in the different items, which may cause the discriminative description to be less effective than it can. One such problem stems from not all objects being labeled in all the images. For example, the object *'wall'* has not been labeled in many images, even though walls exist in all the indoor images that we use. Therefore, if the target image is the only one which has the label *'wall'*, this object will be the first to be described in our discriminative approach, even though it does not actually give any useful information to the subject.

There can also be errors in our color or relationship detector. These would cause more problems for a discriminative description than for a random de-

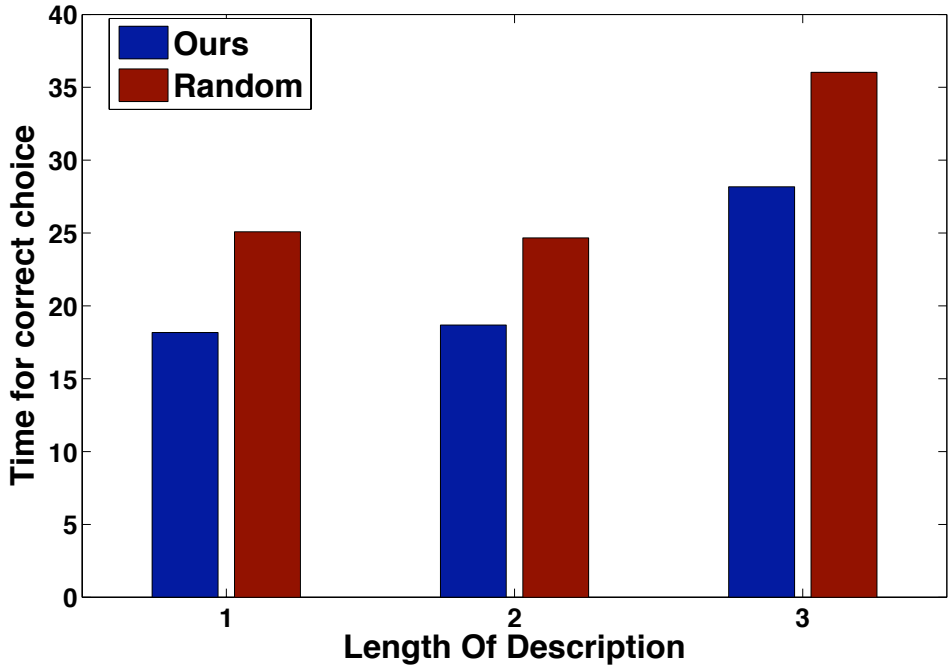

Figure 2.7: Average time to guess correctly. For each description length (up to three) we take only the subjects who have guessed correctly and calculate the average time they spent guessing at that description length.

scription. Since an error in these detectors might create a very unlikely item, there is a high probability that it will be the first to be mentioned in the discriminative description, and might end up throwing the subject off. This is in contrast to the random approach, where this item might not be chosen to be described until later.

In Fig. 2.7, we also plot the average time it took for subjects to guess correctly, for each description length up to 3. From this plot, it is clear that people were able to guess the correct answer given our description 7 seconds quicker on average. This make sense, since if the items describing the image existed in

only one or very few images, the subjects would need less time to choose the correct answer.

2.7.2 Parameter Evalutation

In this section we present the results from our experiments on Amazon's Mechanical Turk. During our study, we had 159 unique workers guessing 90 different sets of target/distractor images (30 for each parameter). For each target/distractor set, we created 4 different description lengths for each parameter setting. Since we examined three values for each parameter, and allowed 4 workers to work on every task, we ended up conducting a total of about $90 \times 4 \times 3 \times 4 = 4320$ tasks.

Since our focus was to test if the saliency measures we are using could improve efficiency, we conducted the following experiment. First, we selected manually images for which we expected these measures to make a difference. This allows us to show that these measures can actually be useful for discriminating between scenes. Second, instead of paying each person a constant sum we pay only \$0.01 per task, but then pay people who guess correctly an extra of \$0.02, thus tripling their reward amount.

Fig. 2.8 presents the results of this experiment. Although it has been conducted on a relatively small set of images, which were chosen specifically for this task, some interesting observations can be made. First, from all three graphs it is clear that all the parameters can be helpful in determining what are the most useful items to describe. This supports our initial assumption, by showing that in the case of visual scenes, mere discriminability will not always produce the

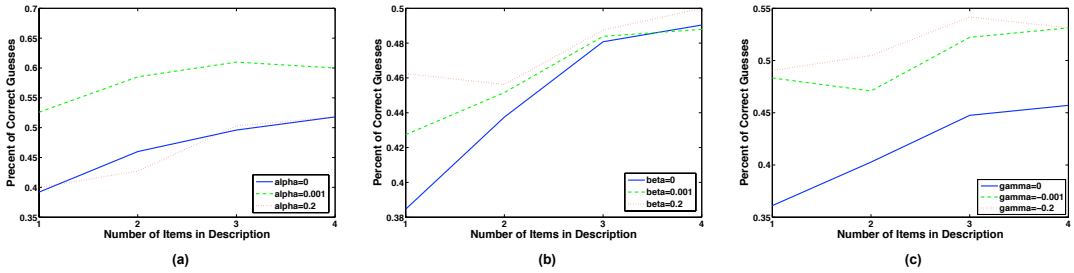

Figure 2.8: The results of our three Amazon Mechanical Turk experiments. In each experiment we examined the effect of one of the parameters and set the other two parameters to 0. (a) The effect of α which is the weight given to the size of the object. (b) The effect of β which is the weight given to the low level saliency of the object as described in the model by [17]. (c) The effect of γ which is the weight given to the centrality of the object.

best results. Each of the three factors seem to provide some benefit to the algorithm.

We examine how different people responded to the same target/distractor set given the different description, and find the ones in which the different saliency parameters made the most difference. Examples of these can be shown in Fig. 2.9. Fig. 2.8 (a) also shows an interesting effect of these parameters: if these end up being too high, they can make the description worse. This is fairly obvious, since the more weight we put on saliency, the more probable it is that a high ranked item might also appear in other images. For example, although a chair is much bigger than an apple, if it appears in many of the distractors it might reduce the probability of a correct guess. Although this effect does not show itself in the other two results, we expect that if we raise the parameter even higher the effect will be the same.

Another interesting observation is the performance increase for the different

Figure 2.9: Image examples of how saliency can assist in discriminating between images. The colors represent different values for the different parameters, while the graph shows the improvement in performance for each parameter for that specific image. (a) On the left of the image there is a small basket above the sink. This is very hard to notice. However, the plant next to the cabinet in the front-right of the image is much easier to see and therefore provides a 25% increase in guessing. (b) There is a cup in the middle of the image. However, since it is clear it has very few edges. Although the outlet is small it has a much higher saliency score and thus provides a 30% increase. (c) Although both the carpet and the curtain only existed in this image out of all the distractors, the curtain is centered, so it provided a 12% increase. (d) Although using the size parameter helps in choosing the carpet over the basket, if it is too high then too much weight is given to the size and it selects a non-discriminative item.

parameters. That is, both size and centrality seem to increase the performance around 15% while our saliency model only gives a 8% increase which reduces to just a few percent for descriptions longer than one item.

2.8 Future Work

Since generating discriminative descriptions for images has never been attempted before, there are many possible extensions to this work. For example, we plan to collect human-generated discriminative descriptions using Amazon's Mechanical Turk. The basic idea would be to use the same data set of labeled images in a similar setting, but instead of requiring the subjects to find the target image, they would be provided with the image, and would need to generate a description. By analyzing the statistics of what they chose to describe, in relation to the objects that appear in the image, we should be able to build a more reliable model.

An additional extension can involve looking at more general descriptions that are not task specific. It has been shown in the past that people tend to name the same objects in an image relatively consistently when not presented with a definite task [35]. It would be interesting to examine how people choose what to describe (not only objects, but relationships and colors as well) given a general task of describing an image, and then try to build a model to replicate that.

There are other properties in the image that we have not examined in this chapter. On the object level, there are many more attributes that can be described. On the scene level, the scene category, lighting, and coloring might be of use. Finally, it may be possible to infer attributes such as actions or feelings

from the image. How to integrate all these details into one coherent description remains an open problem.

CHAPTER 3

REFERRING EXPRESSIONS FOR OBJECTS

3.1 Introduction

Imagine you are at a party with many people, and need to point out one of them to a friend. Because it is impolite to point (and it is difficult to follow the exact pointing direction in a large group), you describe the target person to your friend in words. Most people can naturally decide what information to include in what is known in the Natural Language Processing field as a *referring expression*. For example, in Figure 3.1, we might say: (a) "The man who is not wearing eyeglasses" (b) "The man who is wearing eyeglasses" or (c) "The woman".

The task of generating these expressions requires a balance between the two properties of Grice's Maxim of Quantity [12]. The maxim states:

- Make your contribution as informative as is required.
- Do not make your contribution more informative than is required.

In our context, in which the computer attempts to refer to a single person, we interpret these as follows. First, the description ideally refers to only a single target person in the group such that the listener (guesser) can identify that person. Second, the describer must try to make the description as short as possible.

Although people find this describing task to be easy, it is not trivial for a computer. First, computers must deal with uncertainty. That is, the attribute classifiers the computer uses are known to be noisy and this uncertainty must

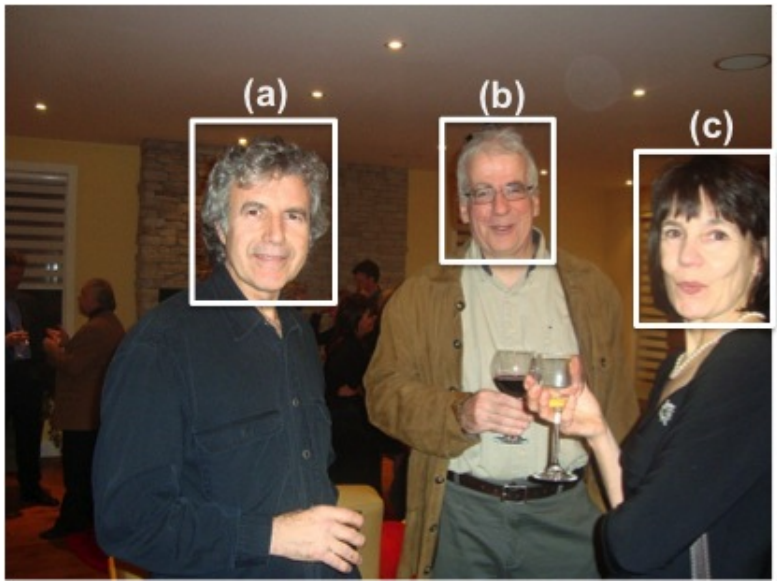

Figure 3.1: In this chapter we introduce an efficient method for choosing a small set of noisy attributes needed to create a description which will refer to only one person in the image. For example, when the target person is person *(b)*, our algorithm produces the description: "Please pick a person whose forehead is fully visible and has eyeglasses"

be considered in an effective model. In addition, given that each person in our image might have many attributes describing him, selecting the smallest set of attributes with which to describe him uniquely is an *NP-hard* problem[6]. For example, a brute-force method is to first try all descriptions with one attribute, then try all descriptions with two attributes and so on. Although this will find the shortest description, the computational complexity is exponential in the number of available attributes. Finally, our attribute vocabulary might not allow us to generate an expression which can refer solely to our intended object. However, since we are using visual data we can incorporate additional information such as absolute location (where the person is in the image), and relative location (who the intended person is standing next to).

This task represents an important part of a broader set of problems which

address generating general descriptions for images. This is evident from the fact that referring expression generation is considered one of the basic building blocks for any natural language generation system [25]. When giving a general description one might be required to refer to specific objects within the scene. For example in Figure 3.1, we might say "The person wearing eyeglasses is the company's president," instead of simply "The person is the company's president." This type of referral is crucial in generating informative image captions. Our algorithm provides a method for selecting which attributes should be mentioned in such a case.

In fact, recent developments in the robotics community provide a direct necessity for these type of expressions. For example, Tellex et al. [36] describe a collaborative scenario in which a robot assembling furniture might need to ask a human for assistance in the form of "Hand me the white table leg". Although they do not use visual data in that work for generating the referring expressions, this type of scenario would require an algorithm similar to ours to perform optimally. In addition, Walter et al. [38] describes a scenario where a robot is learning a semantic map from natural descriptions. In this case the robot might again need to ask the human questions related to a specific room, such as "Is this the office in front of the kitchen". These scenarios will again require a referring expression generation in face of uncertainties and physical locations.

There are also additional practical applications. In security, surveillance cameras and action recognition algorithms can identify suspicious people. A security guard could receive concise verbal descriptions of the suspect to investigate. Both properties of the description are extremely crucial. First, the de-

scription needs to refer only to the suspect to prevent investigating the wrong person. Second, it must not be too long as to confuse the guard or waste his time.

Another application involves navigation systems. Using a front-facing camera on a car and a GPS system, we can develop a system which can provide more intuitive driving directions. For example, instead of saying: "Turn right in 200 feet," it might be more useful to say: "Turn right at the yellow building with the red awning," or even "Follow the green car that just turned right." Although we use our algorithm for describing people, it is is not confined to this specific domain. By employing object detection algorithms, in addition to other attribute classifiers, a general system can be realized.

Our main contributions are: We present the first attempt at generating referring expressions for objects in images. This task has been researched in the NLG community, but had yet to use visual data with actual uncertainties. In addition, we present a novel and computationally efficient method for evaluating the probability that a given description will result in a correct guess from the listener. Using this, we develop a new algorithm for attribute selection which takes into consideration the uncertainty of the classifiers. That is, although we cannot guarantee that the description we compose will describe only the target person, we are able to select attribute combinations for a high probability of this occurring. Finally we evaluate the benefit of including both absolute and relative locations of the objet in the image.

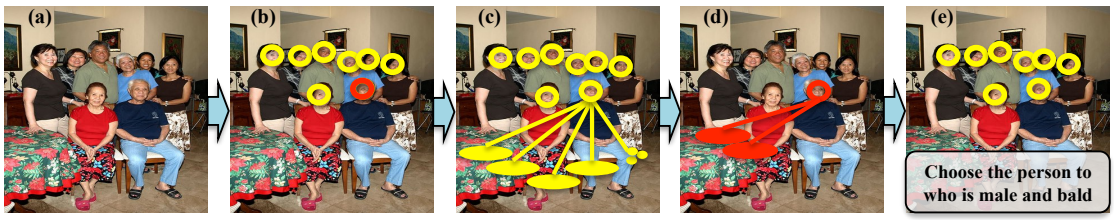

Figure 3.2: An overview of our algorithm. (a) Given an image of a group of people (b) detect all faces and select a random target. (c) For each face run a set of attribute classifiers. (d) Find a small set of attributes which refers to the target face with confidence c (e) Construct a sentence and present to a guesser.

3.1.1 Previous Work

There has been active computational research on referring expression generation in the NLG community for 20 years. Most consider a setup in which there exists a finite object domain D each with attributes A. The goal is to find a subset of attribute-value pairs which is true for the target but false for all other objects in D. We build on this work from a computer vision point-of-view, using actual attribute predictions made from analyzing real images of people.

One of the earliest works include Dale's *Full Brevity* algorithm [4] which finds the shortest solution by exhaustive search. Since this results in an exponential-time algorithm two main extensions were introduced in [6]. The Greedy Heuristic method chooses items iteratively by selecting the attribute which removes the most distractors that have not been ruled out previously until all distractors have been ruled out. The Incremental Algorithm considers an additional ranking based on some internal preference of what a human describer would prefer, in an effort to produce more natural sounding sentences. Our goal is the same (to produce discriminative descriptions), but we consider the confidence scores of real attribute classifiers, and introduce an efficient algo-

rithm for dealing with this uncertainty.

Other extensions to these three main algorithms have been proposed. For example, Krahmer etal propose a graph base approach for referring expression generation [19]. The reason for using this approach is that it allows for relationships between objects to be expressed (for example spatial relationships) in addition to the individual attributes of each object. We use a similar graph in our work.

Horacek proposes an algorithm which deals with conditions of uncertainty [16]. This method is similar to the one we are proposing since it does not rely on the fact that the describer and the listener agree on all attributes. However, our algorithm differs in important ways. First, we provide a method for efficient calculation under uncertain conditions whereas in Horacek's paper the calculation is computationally expensive. In addition, Horacek's definition of the uncertainty causes is heuristic, but we use calculated uncertainties of classifiers. And, in contrast to [16], we provide experimental data to show our algorithm's strength.

Although this is the first attempt at generating referring expressions for objects in images, our work is an extension of previous work researching attribute detection and description generation. For example, Farhadi et al. [7] detect attributes of objects in scene, and use them as a description. The initial description includes all attributes and results in a lengthy description. With no task in mind, they are not able to measure the usefulness of the description. In our work, which is task specific, we are able to select attributes in a smart way, and show the utility of our descriptions.

Attributes improve object classification [23, 29] and search results [18]. For example, Kumar et al. describe in-depth research on nameable attributes for human faces. These attributes can be used for face verification and image retrieval [22], and similarity search [34]. These works all use human-generated attribute feedback to help a computer at its task. In contrast, in our case the computer (not a human) is the one generating descriptive attribute statements, so the emphasis is on selecting attributes, even when the classifier scores are uncertain.

In recent years, attributes have been used to automatically compose descriptions of entire scenes. Although this is different from describing a specific object within a scene, there are similarities. For example, Berg et al. [1] predict what is important to mention in a description of an image by looking at the statistics of previous image and description pairs. They mention a few factors (e.g., size, object type and unusual object-scene pairs) to help predict whether an item will be mentioned in a description.

Both Farhadi et al. [8] and Ordonez et al. [27] find a description from a description database that best fits the image. Gupta et al. [13] use a similar approach, but break descriptions into phrases to realize more flexible results. Kulkarni et al. [21] use a CRF infer objects, attributes and spatial relationships that exist in a scene, and compose all of them into a sentence. The main difference between this line of work and ours is the fact that our description is goal-oriented. That is, prior works focus solely on the information and scores within the scene. In contrast, we consider attribute scores for all objects to describe the target object (person) in a way that discriminates him from others.

Finally, Sadovnik et al. [33] produces referring expressions for entire scenes.

However, our method improves on [33] in major ways. First,[33] ranked various attributes, but did not provide a calculation of how many attributes should be used. In our method, we calculate the necessary description length. Second, we rigorously deal with the uncertainty of the attribute detectors, instead of using a heuristic penalty for low confidence as in [33]. Finally, creating referring expressions for objects in a scene as opposed to entire scenes is more natural and has more practical applications (as described in Sec. 3.1).

3.2 Considering Attribute Uncertaintity

3.2.1 Attribute detection

Although the description algorithm we present is general, we choose to work with people attributes because of the large set of available attributes. Kumar et al. [22] define and provide 73 attribute classifiers via an online service. We retain 35 of the 73 attributes by removing attributes whose classification rate in [22] is less than 80%, and removing attributes which are judged to be subjective (such as attractive woman) or useless for our task (color photo). In the future other attributes can be easily incorporated into this framework such as clothing or location in the image.

Each classifier produces an SVM classification score for each attribute. Since our method requires knowledge about the attribute's likelihood, we normalize these scores. We use the method described in [40] which fits an isotonic function to the validation data. We first collect a validation set for our 35 attributes, and fit the isotonic function using the method described in [2].

Variable Name	Variable Description
n	Number of people
$f \in \{1, 2, \ldots, n\}$	Person to be described
$\mathbf{A}$	Set of binary attributes
$\mathbf{a}^* = [a_1^*, a_2^*, \ldots a_q^*]$ $a_k^* \in \mathbf{A}$	The attributes chosen by the algorithm for description
$\mathbf{v}^* = [v_1^*, v_2^*, \ldots, v_q^*]$ $v_k^* \in \{0, 1\}$	Values chosen by the algorithm for the attributes in a^*
$\mathbf{p_k} = [p_{k1}, p_{k2}, \ldots, p_{kn}]$ $k = 1 \ldots q$ $p_{ki} \in [0, 1]$	Probability of attribute k as calculated by classifier for each person
$\mathbf{x_k} = [x_{k1}, x_{k2}, \ldots, x_{kn}]$ $k = 1 \ldots q$ $x_{ki} \in \{0, 1\}$	Values of attribute k of a^* as seen by the guesser
$\tilde{f} \in \{1, 2, \ldots, n\}$	Guesser's guess
$P_{\tilde{f}} = P(\tilde{f} = f \mid \mathbf{a}^*, \mathbf{v}^*)$	The probability of the guesser guessing correctly
$t = \sum_{i=1}^{n} (x_{ki} == v_k^*)$	Number of faces with correct attribute value

Table 3.1: Variable definitions

As stated in Sec. 3.1 the goal of a referring expression generator is to find a short description that refers to a single object in the scene. In our scenario of uncertain classifiers, our goal is to produce a description that will allow a guesser a high probability of successfully guessing the identity of the target face. Calculating this probability relies on a guesser model which we provide in Sec. 3.2.2. The guesser model defines the strategy used by the listener to guess which face in the image is the one being described.

We then describe how to calculate the probability that the guesser will, in fact, guess the target face given any description within the space of our attributes by considering the uncertainty of the attribute classifiers. First, we explain this calculation when the description has a single attribute (Sec. 3.2.3). Then, we explain the extension to the case when the description contains multiple attributes (Sec. 3.2.4). In both cases, we show that this calculation is polyno-

mial in both the number of faces in the image, and the number of attributes in the description.

Finally, we introduce an algorithm for producing attribute descriptions that meet our goals: having as few attributes as possible, while selecting enough so that that probability of a guesser selecting the the target person will be higher than some threshold (3.2.5). We also provide a worked out example of the attribute selection algorithm in Appendix A.

3.2.2 Guesser's Model

We first define a model that the guesser follows to guess the identity of the target person, given an attribute description. All variables are defined in Table 3.1. Given that he has received a set of attribute-value pairs $(\mathbf{a}^*, \mathbf{v}^*)$, he guesses the target face $\tilde{f}$ according to the following rules:

- If only one person matches all attribute-value pairs guess that person.
- If more than one person matches all attribute-value pairs guess randomly among them.
- If no person matches any attribute-value pairs guess randomly among all people.
- If no person matches all attribute-value pairs, choose randomly among the people who have the most matches.

Given this model, the describer's goal is to maximize $P_{\tilde{f}} = P(\tilde{f} = f|\mathbf{a}^*, \mathbf{v}^*)$, the probability that the guesser correctly identifies the target, given the description.

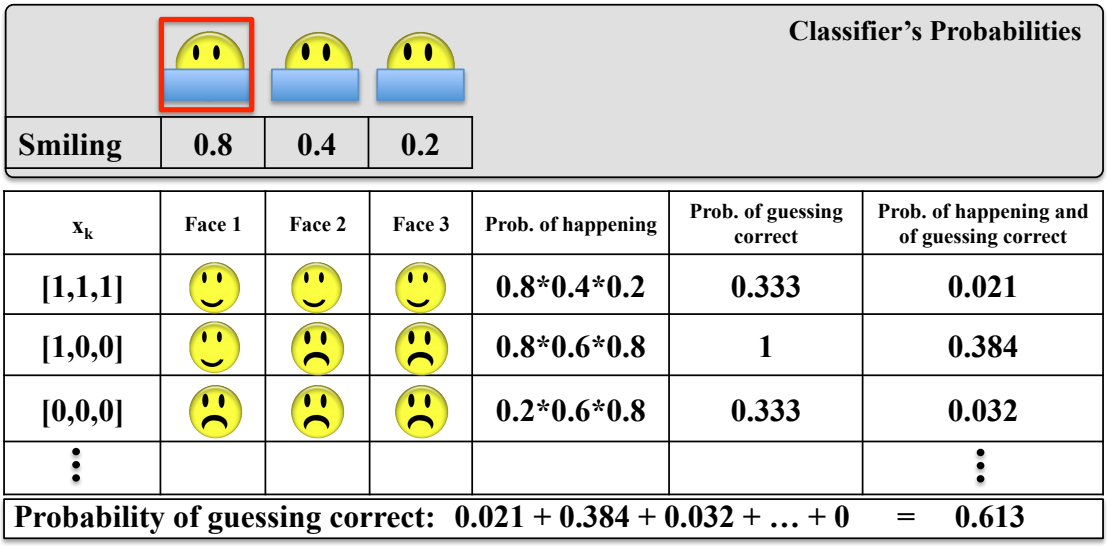

Figure 3.3: An illustration calculating the probability of guessing correctly using one attribute ("The person is smiling") for an image with three people. The true identity of the target person (marked with a red rectangle) is known to the algorithm as well as the attribute confidence for each face. Each face is actually smiling or not (the true state is unknown to the algorithm), represented with the blind over each mouth. To find the probability of the guesser's success, each of the eight possible configurations of smiling faces is considered. We introduce a polynomial-time algorithm for computing this probability.

Following Grice's Maxim of Quantity we also wish to create a short description. Therefore, we choose to explore descriptions that minimize the number of attributes $|a^*|$ such that $P_{\tilde{f}} > c$, where c is some confidence level.

To show how $P_{\tilde{f}}$ is calculated we first present the single attribute case, and then extend to multiple attributes.

3.2.3 Single Attribute

Consider the case where a "smile detector" is applied to an image containing three faces, and we refer to face 1 as "the smiling face" (Figure 3.3). What is the probability that a guesser will be correct? To compute this, we must consider the fact that our smile detector is never certain, but instead, reports confidences of observing a smile on each face. The confidence associated with each score represents the probability that each face actually has a smile or not. The actual joint distribution of smiling faces in the image has eight possibilities over the three faces (2^3). For each of these eight possible arrangements, the probability that the guessing strategy leads to a correct guess can be computed. Naïvely, by applying total probability, the overall probability of guesser success is the sum of the probability that each of these eight smile cases occur, times the probability of guesser success in each case.

We now formalize our algorithm. Here, for simplicity of notation, the description is comprised of positive attributes (e.g., "the smiling face"), but we also consider negative attributes (e.g., "the face that is not smiling") by taking the compliment of the attribute probability scores for each face. The probability of each possible $\mathbf{x_k}$ occurring is:

$$P(\mathbf{x_k}) = \prod_{i=1}^{n} (x_{ki}p_{ki} + (1 - x_{ki})(1 - p_{ki})) \tag{3.1}$$

For each $\mathbf{x_k}$ and attribute-value pair (a_k^*, v_k^*) we compute the probability of the guesser guessing correctly using the guesser model:

$$P(\tilde{f} = f|\mathbf{x_k}, a_k^*, v_k^*) = \begin{cases} \frac{1}{n} & \text{if } t = 0 \\ 0 & \text{if } x_{kf} = 0 \ \& \ t > 0 \\ \frac{1}{t} & \text{otherwise} \end{cases} \tag{3.2}$$

Therefore, we calculate the total probability of a correct guess given a single attribute by summing over all (2^n) configurations of the attribute over the faces in the image as:

$$P_{\tilde{f}} = \sum_{\mathbf{x_k}} P(\tilde{f} = f|\mathbf{x_k}, a_k^*, v_k^*)P(\mathbf{x_k}) \tag{3.3}$$

In Eq. 3.3, we sum over all possible $\mathbf{x_k}$ which is exponential in the number of faces n and computationally expensive. Since the images in our dataset contain many faces, it is intractable. However, we notice that $P_{\tilde{f}}$ depends only on the number of faces t that satisfy the attribute, given that the target face does. We can rewrite Eq. 3.3 as:

$$P_{\tilde{f}} = \frac{1}{n}P(t = 0) + 0 + \sum_{\mathbf{x_k}|x_{kf}=1} \frac{1}{t}P(\mathbf{x_k}) \tag{3.4}$$

Where each of the three terms in the sum refer to the three terms in Eq. 3.2 respectively. Finally, we notice that t is actually a Poisson-Binomial random variable whose PMF (probability mass function) can be computed in time polynomial with the number of faces. A Poisson-Binomial distribution is the distribution of the sum of independent Bernoulli trials where the parameter p can vary for each trial (as opposed to the Binomial distribution). We can calculate the PMF efficiently by convolving the Bernouli PMF's [9]. In our case, the parameters of the random variable are p_k . We can therefore rewrite Eq. 3.4 as:

$$P_{\tilde{f}} = \frac{1}{n}P(t = 0) + 0 + p_{kf} \sum_{t=1}^{n} \frac{1}{t}P(t|x_{kf} = 1) \tag{3.5}$$

Since inside the summation we only care about cases in which $x_{kf} = 1$ we set the Poisson-Binomial parameter for face f to 1 and then compute the PMF of t. Eq. 3.5 provides a way to calculate the value of Eq. 3.3 exactly while avoiding the summation over all possible $\mathbf{x_k}$. We can now compute $P_{\tilde{f}}$, the probability that the guesser will succeed, in time ploynomial with the number of faces.

	Face 1	Face 2	Face 3	Face 4
Hat	**0.90**	**0.20**	**0.80**	**0.10**
Beard	**0.60**	**0.60**	**0.80**	**0.90**
White	**0.30**	**0.40**	**0.90**	**0.50**

	Face 1	Face 2	Face 3	Face 4
0 Att.	**0.03**	**0.19**	**0.00**	**0.05**
1 Att.	**0.31**	**0.46**	**0.07**	**0.45**
2 Att.	**0.50**	**0.30**	**0.35**	**0.45**
3 Att.	**0.16**	**0.05**	**0.58**	**0.05**

Figure 3.4: An example of transforming the table of p_{ki} into the 4 PMF's of y_i (one per column). In Eq. 3.8, j iterates through the different rows and normalizes accordingly.

Using Eq. 3.5 we can find, from a pool of available attributes, the single best attribute to describe the target face (the a_k^*, v_k^* that maximizes $P_{\tilde{f}}$). Extending this strategy to multi-attribute descriptions is not trivial. One greedy algorithm for producing a multi-attribute description is to order all available attributes by $P_{\tilde{f}}$, and choose the top m. However, this could yield redundant attributes. For example, imagine a group photo with two people who both have glasses and are senior, one of whom is our target. The attribute-value pairs *has glasses* and *is senior* may be the top two with the greatest $P_{\tilde{f}}$. However, mentioning both attributes is useless, because they do not contain new information. What is actually needed is a method of evaluating the guesser success rate with a multi-attribute description.

3.2.4 Multiple Attributes

We introduce a new random variable y_i, the number of attributes of face i which correctly match the description $(\mathbf{a}^*, \mathbf{v}^*)$.

$$y_i = \sum_{j=1}^{q} x_{ji} == v_j^* \tag{3.6}$$

In this work we consider all attributes to be independent. Therefore, y_i is also a Poisson-Binomial random variable whose parameters are $p_{ji} \mid j = \{1, 2 \dots q\}$ (as shown in Figure 3.4). We expand the definition of t from our single attribute example. Whereas previously it signified the number of faces with the correct value for a single attribute, t_j now signifies the number of faces with exactly j matching attributes.

$$t_j = \sum_{i=1}^{n} y_i == j \tag{3.7}$$

Using these random variables we efficiently calculate the guesser's success given multiple attributes. The basic idea is to look at the case when the target face has j correct attributes and no other face has more than j attributes correct (if any other face does the probability of guessing correctly is zero), and then perform Eq. 3.5 using t_j where our new p values are the jth row of Figure 3.4 normalized by the sum of rows $0 - j$. Summing over all values of j gives us the following equation:

$$\begin{aligned} P_{\tilde{f}} = \sum_{j=1}^{q} \sum_{t_j=1}^{n} \Big(& \frac{1}{t_j} p(t_j | y_f = j, y_i \leq j \ \forall i) \\ & \times p(y_f = j | y_i \leq j \ \forall i) p(y_i \leq j \ \forall i) \Big) \end{aligned} \tag{3.8}$$

3.2.5 Guesser-Based Attribute Selection

We perform attribute selection in a similar fashion to the Greedy Heuristic Method. The algorithm's pseudo code is shown in Algorithm 1. This is a greedy method in which in each step we select the best attribute-value pair to add to our current solution, which gives us the highest combined probability of guessing correctly given our selection from the previous step (evaluated with Eq. 3.8).

Algorithm 1: Attribute selection algorithm

Data: c, A, f
Result: a^*, v^*
$a^* \leftarrow \emptyset$;
$curr\_conf \leftarrow 0$;
while $(curr\_conf < c)$ **do**
  for *each* $A_i \notin a^*$ **do**
    $tmp\_A \leftarrow a^* \cup A_i$;
    for *each* tmp_v **do**
      calculate $p = P(\tilde{f} = f | tmp\_A, tmp\_v)$;
      if $p > curr\_conf$ **then**
        $curr\_conf \leftarrow p$;
        $curr\_best \leftarrow (tmp\_A, tmp\_v)$
      end
    end
  end
  $(a^*, v^*) \leftarrow curr\_best$
end

Once we have a set of attributes we construct a sentence. Since the main focus of this chapter is on the selection method we create a simple template model to build the sentences.

3.3 Considering absolute location

Assuming that the speaker and the viewer are seeing the scene from the same point of view (which is always true in the case of images), we can use the absolute location of the object within the scene. A trivial way to add this information is by considering location as just an additional attribute. We can do so by fitting a sigmoid function in both dimensions of the image. The vertical sigmoid represents $p(right) = 1 - p(left)$, while the horizontal sigmoid represents $p(front) = 1 - p(back)$. Since we are focusing on faces, we reshape the horizontal and vertical sigmoids to lie between the leftmost and rightmost person and the topmost and the bottommost person respectively. Fig. 3.5 shows an illustration

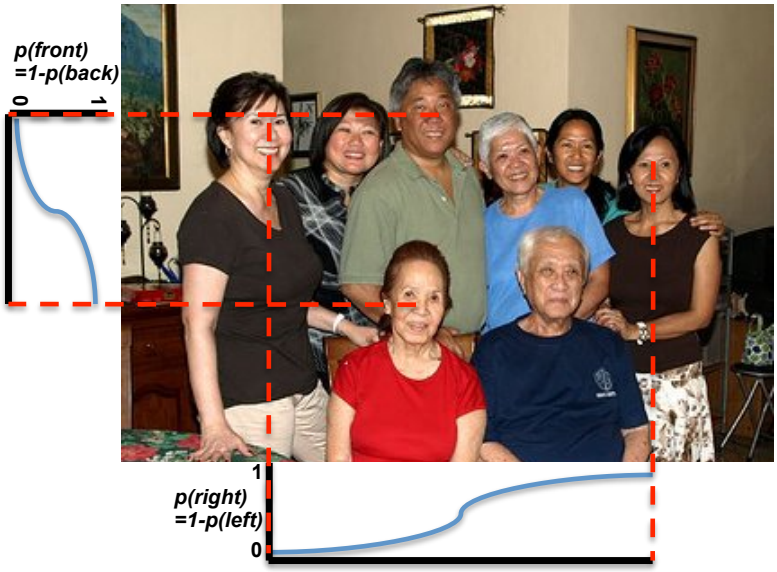

Figure 3.5: Fitting a logistic function to the location based attributes. .

of these sigmoids. Once we have these probabilities we can simply treat them as two additional attributes and perform the original algorithm as described in Sec. 3.2.2

There is an additional way to use absolute location in an image which is very commonly used in newspaper captions. Instead of using attributes it is possible to simply use the absolute location by stating the row number, and position within that row of the target person. Although this method is useful in many cases we predict it would not perform as well as the attribute based method for two main reasons:

1. Our row detection algorithm is not perfect, and even a small error can create a completely wrong description.
2. In some images people are not organized in a row like fashion and therefore these descriptions will be confusing.

Figure 3.6: An example of face rows detected in the image.

In order to verify this we construct a set of row based descriptions. We first use the algorithm by [10] to divide the people in the image in to rows. First a full graph over the faces is built. Each edge weight is a function (which is learned from ground truth row segmented training data) of the difference in position and size of the two connecting vertices. Once the graph is built it can be divided into subgraphs, which represent rows, by iteratively performing the min cut algorithm. An example of detected rows is shown in Fig. 3.6. We then create the following description based on the target's face position:

> Please choose the person who is the ^{n}th person from the left/right on the ^{m}th row from the top/bottom

Where, if we start counting rows from the top, and position from the left:

$$n = min(pos\_in\_row, people\_in\_row - pos\_in\_row)$$

$$m = min(row\_number, num\_of\_rows - row\_number)$$

And we choose the left/right or front/back keywords depending on which was variable was returned by the $min()$ function. The reason we use the $min()$

function is to ensure that we are using the simplest description possible, and in addition minimizing the potential for mistakes. For example, in Fig. 3.2, it would make more sense to say *"The first person on the right on the first row from the bottom"* vs. *"The second person from the left on the second row from the top"*.

When examining both types of location based descriptions we observe that both have their own benefits (see Fig. 3.7). That is, in cases where people are organized in a row like fashion, and the row detector successfully detected the rows, using a row based description can lead to more accurate results. However, in other cases it might be detrimental to use these descriptions, and an attribute based description would perform better. We therefore try to develop a method which, given an image with a target person and both types of descriptions (attribute based and row based) attempts to select the best description and present it to the user. We formulate this problem as a binary classification problem, where the two classes are:

- Row based description
- Attribute based description

We examined many possible input features for our experiments based on the image itself (GIST feature vector [26]), the row detector (number of rows, graph cut score), the attribute based description (length of description, description confidence, number of attributes used, which attributes were used), and row description (number of people in target's row, target row number, target column number). We observed the best results when using these three features: description confidence, number of rows in image and length of description. We then examine the guessing accuracy when selecting the best description to present to

Figure 3.7: Two images with row based descriptions vs. attribute based descriptions. The examples clearly show that for some faces row based descriptions would be useful while for others they would not. (Description in green is assessed to be a better referring expression)

the user based on the classifier.

3.4 Considering Relative Location

Given our attribute vocabulary, a certain person might not have enough distinctive attributes to separate him from others in the group. Therefore, we wish to be able to refer to this person by referring to people around him. However, deciding who is standing next to whom is not a trivial task. We again use the work of Gallagher et al. [10] which divides all the faces in the image into rows.

We use this information to define faces who have a common edge in a row as neighbors. This gives us the "to the left of" and "to the right of" relationships. Since in [10] faces can be labeled as in the same row even though they are far

apart, we add an additional constraint which normalizes the distance between every two faces in a row by the size of the face, and removes edges where the normalized size is greater than some threshold t. This prevents distant people from being considered neighbors.

We now can use this neighbor information to assist our algorithm. We do this by setting an upper limit on the number of attributes used. If the algorithm fails to reach desired confidence , we re-run the algorithm using the neighbor's attributes as well. It should be emphasized that when using a neighbor we examine both sets of attributes jointly (that is, our attribute set is doubled). This allows us to create descriptions such as "The person with the glasses to left of the person with the beard".

3.4.1 Considering Relative attributes

So far we have only examined the use of binary attributes. However, when examining the way people use attributes in natural language they do not solely use them in their binary form, but might also use them in there relative form. That is, although we can say if a person is smiling or not, we can also compare two people who are smiling to decide which one is smiling more. In this section we attempt to introduce relative attributes into our framework and examine the benefit of using them.

Relative attributes were first introduced to the computer vision community in [28]. In there work they proposed modeling each attribute as a ranked list. When predicting an attribute, the goal is to rank it correctly amongst all other predictions. Therefore, they use a rankSVM which optimizes for this problem

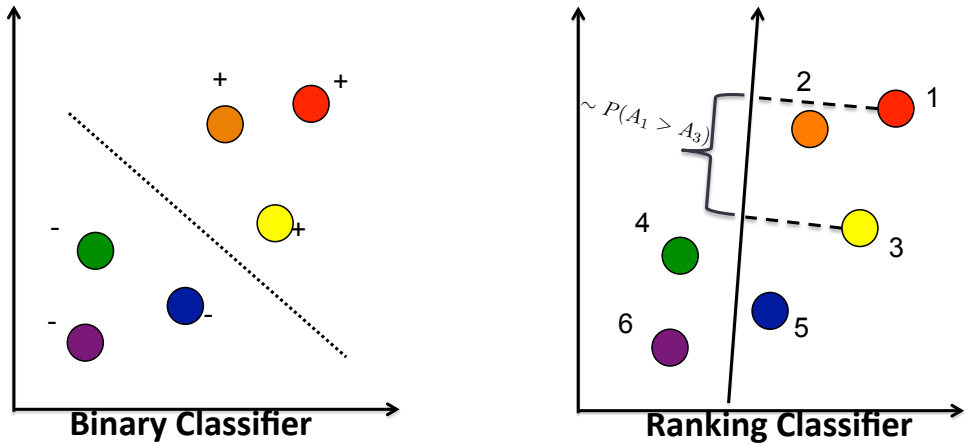

Figure 3.8: A comparison of a binary SVM and Rank-SVM. In our work we normalize the difference obtained by Rank-SVM to a probability that one person has more of an attribute than another face.

exactly. The output of the rankSVM is a score which when compared to other scores can be ranked accordingly.

We propose introducing relative attributes into the referring expression framework by utilizing superlatives such as "most smiling" or "least smiling". For each relative attribute a we add two attributes to our original framework

- Most a
- Least a

We then need to calculate the probabilities of these two new attributes for all faces. We do so using the following formula:

$$P(sup) = \prod_{\forall j | j \neq i} P(att_i > att_j)$$

In order to be able to calculate this probability we first need a way for calculating $P(att_i > att_j)$ for any two faces i, j. We do so by first training a rankSVM as is done in [28] which allows training on pairwise data that includes pairs that are judged to be equal. We use the face pyramid features described in [14]. This a 21504 dimension feature vector which is composed of densely sampled SIFT descriptors at each of the 21 spatial grid cells on a 3-level pyramid. The SIFT descriptors are first encoded using Locality-constrained Linear Coding (LLC) to a 1024-dimension vector and then concatenated to a 21504 length vector. This algorithm returns a weight vector w. By projecting each face's features onto it and taking the difference, the relative strength of the attribute between the pair is found.

In this work we normalize $wx_i - wx_j$ to $P(att_i > att_j)$ as shown in Fig. 3.8. After training the rankSVM we use a cross validation dataset to fit an isotonic function to the normalized difference $wx_i - wx_j$. This allows us to calculate $P(sup)$ and fill the table with two additional superlative attributes for each relative attribute trained. Once these attributes are added we can simply use the same algorithm introduced in Chapter 3 to generate our referring expression.

3.5 Experiments and Results

To evaluate our algorithm we run experiments on AMT. Workers view an image with all detected faces marked with a square and a textual description, and ask them to select who is being referred to. The selection is done by clicking on a face. Each worker performs a random set of ten image-description pairs with one guess each. We encourage the workers to guess correctly by offering a

monetary bonus to the top guessers.

We use images from the Images Of Groups Dataset [11] that contain at least 8 people. This provides us a total of 288 pictures. The face detector detects 87% of the correct faces with 89% accuracy for an average of 11.4 faces per image (random guessing would achieve an average of 0.099). Out of the 3282 faces we randomly select 400 for our experiment.

3.5.1 Computer Baselines

We compare the guessing accuracy for descriptions created using the following methods:

1. **Confident:** Compose the description from the n most confident attributes. This baseline completely ignores other faces in the image.
2. **Top_used:** After running the algorithm on the dataset, we select the n top used attributes throughout the whole set. The top 5 attributes are: gender, teeth visible, eyeglasses, fully visible forehead and black hair.
3. **Full_greedy:** We rank the attributes using the value of Eq. 3.5, skipping the method introduced in Sec. 3.2.4, and use the top n to compose the description.
4. **GBM:** Guesser Based Model. Our full algorithm without neighbors.

We create 2000 descriptions for the 400 faces (1 for each method). We have 3 separate AMT workers guess each, for a total of 6000 guesses. We set our confidence level c to 0.9 and the maximum number of attributes to 5. For faces

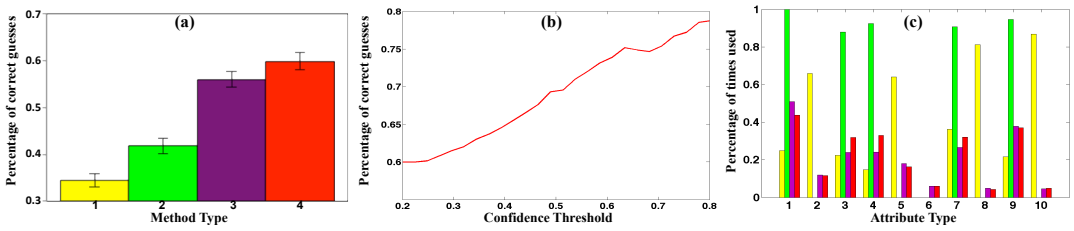

Figure 3.9: Our results from the computer baseline experiment (Sec. 3.5.1). (a) Guessing accuracies for the five methods introduced in Sec. 3.5.1. 1. confident 2. top_used 3. Full_greedy 4. GBM (b) Accuracy results of GBM as we increase the minimum threshold, by looking at descriptions whose confidence level as calculated in Eq. 3.8 are higher than it. (c) The percentage of descriptions (methods 1-4) an attribute was used in for a select set of attributes. The attributes are: (1) Gender (2) White (3) Black hair (4) Eyeglasses (5) Smiling (6) Chubby (7) Fully visible forehead (8) Eyes open (9) Teeth not visible (10) Beard

which do not reach confidence level c, we use the description with the highest score with at most 5 attributes. For the rest of the algorithms, n is the number of attributes selected by GBM. Results are presented in Figure 3.9. We also show description examples in Figure 3.10.

Examining the results, it is interesting that using the most confident attributes actually performs the worst, even worse than simply describing a constant set of attributes as in Top_used (P=0.0022). This shows that an attribute classifier score, by itself, is not enough information to construct an effective description for our task. Figure 3.9c hints at the reason for this. The attributes the classifier tends to be certain about are ones which are not useful for our task since they tend to be true for many people. For example, the *eyes open* attribute (8 in Figure 3.9c) is used in around 80% of the confident descriptions. However, this is usually not useful since most people have their eyes open. This fact is strengthened by the low usage of this attribute by the other methods.

Figure 3.10: Examples of our GBM algorithm along with the calculated confidence and the actual accuracy received from AMT. The left two are examples where our algorithm correctly estimates the confidence (approximately). The right two examples are failure cases: A misclassified target attribute (no hat on target) and a misclassified distractor attribute (additional bearded person in the image).

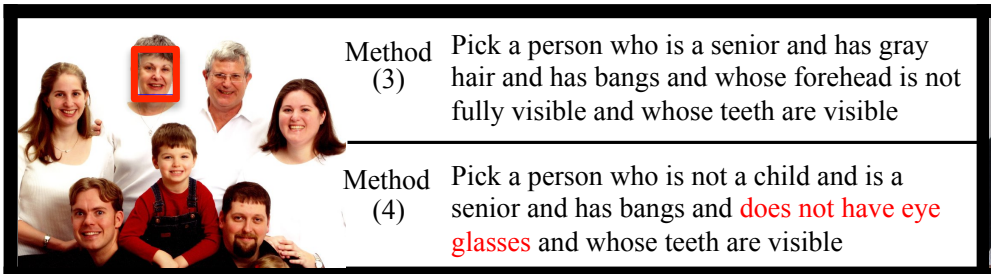

Figure 3.11: Examples of the different descriptions created using Full_greedy vs GBM, and the accuracy achieved in our collected results. The GBM method realized that mentioning *gray hair* after *is senior* is unnecessary and managed to choose a more important .

The need to select attributes in a manner that takes into account the other faces in the image is clear from the improved performance when using our selection algorithms. Our Full_greedy approach reaches an accuracy of 56%. The additional 4% achieved when using GBM (P=0.0131) shows the improvement gained using the methods described in Sec. 3.2.4, which prevent mentioning redundant attributes (See Figure 3.11 for an example).

It is also interesting to investigate how guesser accuracy changes as we change the confidence threshold (Figure 3.9b). Since many of the faces in our algorithm did not reach the necessary confidence, the average confidence of the descriptions is 0.6484 which gives us 60% correct human guesses. It is important to mention that the average description confidence for the entire dataset (all 3282 faces) was 0.6736, which means that our random selection of 400 provided a representative sample of our dataset. However, Figure 3.9b shows that as we increase the minimum confidence, and look only at the descriptions which are above it we can achieve much higher human guessing accuracy. This validates the meaningfulness of our confidence score. In addition, this shows another strength of using GBM since the Full_greedy approach does not present a simple way of calculating this confidence.

3.5.2 Human Describers

We also compare our results using computer descriptions with that of a human describer. In an additional AMT job, workers select attribute-value pairs that best refer to the target person. We reduce the number of attributes to 20 (to simplify the task), and present three radio buttons for each attribute: *not needed*, *yes*, *no*. This is exactly analogous to the computer algorithm and therefore the results are easily comparable. Workers select the fewest attributes that separate the target person from the rest of the group (just as our algorithm does). To encourage workers, we promise a bonus to those whose descriptions give the best guessing probability. We collected 1000 descriptions from 100 separate workers.

Once we have collected all the descriptions given by the workers we create

a new guessing task as described in Sec. 3.5.1. We compare the descriptions created by humans to descriptions created by GBM using the same 20 attributes as given to the user. For this comparison we only use descriptions whose confidence is above 0.7. The descriptions created from the human selection are presented to the guesser in the exact the same format as the computer's. The guesser is never informed of the source of the descriptions (human or computer).

Accuracies from the human and computer descriptions are 76% and 77% respectively. This result validates our model, matching human performance when it attains high confidence of guesser success.

Other interesting observations include that humans tend to use gender much more often than any other attribute (about 70% of the descriptions included gender), while this is not true for the computer algorithm. Even in situations where gender is not necessarily needed, humans still tend to mention it. In addition, humans tend to choose more positive attributes rather than negative ones. In fact, of the 19 attributes (excluding gender since there is no negative for this attribute) 18 were mentioned more often positive than negative. In contrast, for 6 of the 19 attributes, our algorithm mentions the negative attributes more often.

3.5.3 Absolute Location Results

We create two additional types of descriptions as is described in Sec. 3.3:

1. **only_rows** strictly using the exact row position information

2. **GBM_location** Adding absolute location in image as two additional attributes.

When testing the guessing accuracy of these two types on the same 400 faces we tested in Sec. 3.5.1, we achieve 52.0% and 65.7% for only_rows and GBM_location respectively. Compared with the results from Fig. 3.9, we see that only_rows performed worse than our GBM algorithm. As we predicted in section 3.3, the main problems were the errors in row detection, in addition to many images not having row like structures. However, when examining these results we observe that for some images this method works very well and can produce clearer descriptions than the attribute based description.

These results additionally show how adding location as an attribute increases our results. In fact once added, the location based attributes appear in about 80% of the description. These attributes are by far the most used (the next most used attribute, gender, is only used in 30% of the descriptions). The reason for this is mainly since these attributes have more confident probabilities than the other attributes, and are usually discriminative in a sense that they can rule out at least 50% of the people in the image.

Since we observe the only_row method provides good and simple descriptions in many cases, we predict that it should be able to complement our attribute based description. In fact, if an there was an oracle which could choose for each image the type of description (either only_rows or GBM_location) which is best suited for it, we could reach 77.5% accuracy which is a significant improvement over simply using GBM_location. When using the classifier as described in Sec. 3.3 we manage to increase the guessing rate to 67.2%.

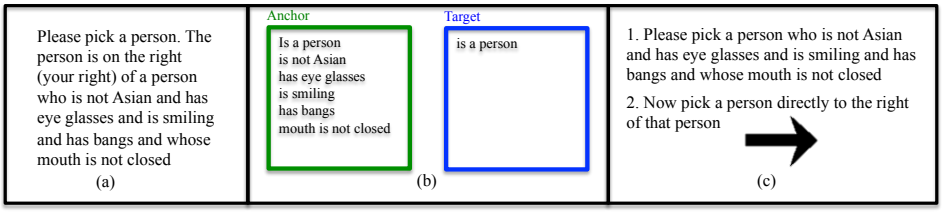

Figure 3.12: Examples of 3 different ways we presented our descriptions. (a) Text: an exclusively textual description as in [31]. (b) Graphical: Our graphical representation (c) Two-Step: Our two step presentation. The second part is only shown after the first part was completed.

3.5.4 Relative Location Results

We create an additional type of description GBM_neighbors based on the algorithm in Sec. 3.4 for the same 400 faces we have used in Sec. 3.5.1, and run the same experiment. However, although GBM_neighbors, which allows using neighbors, had a higher average predicted confidence level (0.82) than GBM (0.65), it produced lower results (52% vs. 59% respectively).

We hypothesized two main reasons why the results achieved by GBM_neighbors were worse. Our first hypothesis was that although the information in the description should yield higher guessing results, the sentence itself was unclear, and was presented in a way that confused the guesser. For example, the user might have been confused about the direction (right vs. left), or confused about who to select (anchor face vs. target face).

In order to test our first hypothesis, we created a new presentation using a graphical diagram instead of the textual description. An image with two squares was presented to the user (Fig. 3.12(b)), one labeled target and the other labeled anchor, and within each square the relevant attributes were listed. We

	Graphical GBM_neighbors		Graphical GBM_neighbors*		Two-step GBM_neighbors*	
	True Anchor	True Target	True Anchor	True Target	True Anchor	True Target
Guessed Anchor	42.0%	9.5%	46.8%	7.6%	64.3%	3.1%
Guessed Target	10.8%	30.6%	7.8%	31.1%	3.3%	55.7%
Sum	52.8%	40.1%	54.6%	38.7%	67.6%	58.8%
	(a)		(b)		(c)	

Table 3.2: Results of our three different experiments as described in Sec. 3.5. (a) and (b) use the presentation method as shown in Fig. 3.12(b), while (c) uses the presentation method as shown in Fig. 3.12(c). The last row is the sum of the first two, and signifies the total percentage of people who chose the true target/anchor as one of their choices.

GBM	Pick a person who has bangs and whose forehead is not fully visible and whose teeth are visible and is wearing lipstick and is not black	1/3	Pick a person who does not have black hair and does not have eye glasses and is chubby and is smiling and whose teeth are visible	1/3	Please pick a person who is a male and is in their youth and has black hair and does not have eye glasses and does not have a mustache	1/3
GBM_ Neighbors*	Pick a person. The person is on the left (your left) of a person who has a mustache and has a beard and whose teeth are not visible and is not black	4/4	Pick a person. The person is on the right (your right) of a person who is a male and has black hair and whose forehead is not fully visible and does not have a mustache and whose teeth are visible	4/4	Pick a person. The person is on the left (your left) of a person who is a child and is not middle aged and has black hair and whose mouth is closed and whose teeth are not visible	4/4

Figure 3.13: Examples of the different descriptions created using GBM vs GBM_neighbors*, and the accuracy achieved in our collected results. In these examples it is clear to see that since it is hard to differentiate the target person from the distractors, using a neighbor anchor face clearly simplifies the task.

then required the user to select both the anchor and the target face. We believed that this graphical representation would solve the confusion of left and right, and in addition, by forcing the user to select the anchor, we could better analyze the error types.

Our second hypothesis was that people were having a hard time finding the correct person since both the target and the anchor face were described as a unique pair. That is, when choosing the attributes to include in the description,

we allow the algorithm to try ones from both the target and anchor face. Therefore, although the description refers to this pair with high confidence, it requires a comparison to all other pairs which might prove too difficult for the average Mechanical Turk user.

In order to test our second hypothesis, we created a new type of description: GBM_neighbors*. In this model, if we cannot create a description with a confidence above the threshold for just the target we look at the target's neighbors individually, and choose the description with the highest confidence. That is, these descriptions will only include attributes from one anchor person as opposed to GBM_neighbors which allowed selecting attributes from both. If a neighbor's description has a higher confidence, we simply request the user to select the person to the left/right of the described anchor person. Although this model produces lower confidences than GBM_neighbors (0.77 vs. 0.82), it creates a description of a single person which, according to our second hypothesis, is clearer.

Using GBM_neighbors* allowed us to try a different presentation. Since the anchor face is the only one described with attributes, the user could guess iteratively. First the user is asked to select the anchor face only. This task is the same as the testing performed on our regular GBM model. Once a face is selected the user is prompted to select an additional face to the left/right of the first selected face. In order to clarify the direction we present an arrow (Fig. 3.12(c)).

For this experiment we use the same images from Sec. 3.5.1. However, since our focus is on the differences between these algorithms, we only run our experiments on the 165 faces for which neighbors were used. As described in Sec. 3.4 if the algorithm reaches the minimum confidence we do not try to use the neigh-

bors, and therefore, on the additional 235 faces, all three algorithms produced the same description.

When we use the original GBM and GBM_neighbors descriptions on this subset of the dataset we achieve 41.47% and 36.6% accuracy respectively. These results are inline with the results on the entire dataset which show that using GBM_neighbors decreases the guessing accuracy. The lower overall performance is expected since we are only looking at the 165 faces for which the confidence score was below the threshold for GBM.

We next tried our graphical representation as shown in Fig. 3.12(b). In these experiments we asked the users to select the anchor face as well, and so had greater insight into errors. Table 3.2(a) presents a confusion matrix of guessed/true anchor/target faces. The columns do not add up to 1 since many faces selected were neither the target nor the anchor.

When looking at the target guessing accuracy (30.6%), we observed an actual decrease form the textual presentation of the GBM_neighbors description. However, when adding up the number of true targets guessed as anchors, we observe an accuracy increase (40.1%), indicating confusion about whether the worker should select the anchor face, or the target face.

Our next experiment presented descriptions created by GBM_neighbors* in the same graphical format 3.12(b). Since the description given to the anchor face selected by the algorithm will definitely have a higher confidence, we predicted that at least the guessing rate for the true anchor will be higher that that for the target of the GBM algorithm. Results are presented in Table 3.2(b) . Although the guessing rate for the true anchor had improved as expected (46.8%), the

target guessing accuracy remained comparable to GBM_neighbors.

This motivated our two-step presentation method (Fig. 3.12(c)). We reasoned that if people are able to guess the anchor face with higher accuracy, then the main problem was still with understanding where the target face is in relation to it. This new presentation method breaks the task into two steps and clarifies the exact direction in which the additional face needs to be chosen. Table 3.2(c) presents the results of this experiment. It is important to note that this type of iterative description would not work for GBM_neighbors, since that method describes the pair jointly and cannot be reduced to two independent selection tasks.

As predicted, this final combination of GBM_neighbors in addition to our two step presentation method performs the best on both target and anchor faces. The higher accuracy for the anchor face vs. the target face is to be expected since getting the anchor correct does not guarantee guessing the target even if the direction is clear (it can be ambiguous who is exactly to the right of a person).

3.5.5 Relative Attributes Results

We trained 5 relative attributes, thus adding a total of 10 additional superlative attributes to the original 35 used in Chapter 3. The attributes we trained were:

- more/less smiling
- more/less teeth visible
- more/less bald
- more/less bearded

- more/less masculine

We then re-generated expressions for our original database of 400 people. However, superlatives were only used rarely (20/400). This makes sense since the probability of choosing a person which has the most of some attribute is low in images with large groups of people. We therefore create a new database with 175 images for which superlatives are used. We then repeat the experiments as described in Chapter 3 for two sets of descriptions, namely: with and without superlatives.

Our results are shown in Fig. 3.14. As is clearly shown the introduction of relative attributes increases the guessing rate significantly. However, since the general guessing rate is much lower than that of our original algorithm (45% vs 60%), it shows that the relative attributes became useful in cases where the binary attributes could only generate a weak description. Fig. 3.15 shows two examples where superlative attributes helped (green background) and two failure examples in which adding these attributes reduced the guessing rate.

For the success cases it is clear to see the benefit gained from adding these superlative attributes. In both cases the binary attribute version of the attribute used in relative form (is smiling/teeth are visible respectively) were not useful for the description since there are other people in the image who have this attribute. However, since these people clearly exhibit this attribute the most in their respective images, these attributes increase the guessing chance vs. the other attributes chosen when only using binary attributes (forehead is fully visible/teeth are visible).

That said the failure examples show that since relative attributes are "softer" than binary ones, they are not always the most useful. Although arguably the people in their respective images are the most smiling/have their teeth most visible, it is somewhat hard to judge that, and can lead people to make mistakes. These come in contrast to does not have bangs/has gray hair which are used in the binary case which are clearly much easier to judge in these examples.

3.6 Conclusion

We have introduced a new approach for solving the novel task of producing a referring expression for a person in an image. We compute a confidence score for each description, based on a novel, efficient method for calculating the score. In addition, we show how the use of both relative and absolute location can help in generating better referring expressions. Finally, we demonstrate the effectiveness of our attribute selection algorithm, comparable even to constrained human-made descriptions.

We believe there are many exciting future directions for this work. First, more can be learned from our human describers and guessers. Our guesser model still does not completely mimic a human because it does not consider factors such as saliency or relative attributes. By examining the human descriptions and guesses, we may learn a better model for the human guesser and redesign our algorithm for referring expression generation.

In addition, this work can be extended to consider back-and-forth conversations between humans and computers. That is, if the referring expression isn't clear, what questions can the guesser ask to clarify her understanding? This

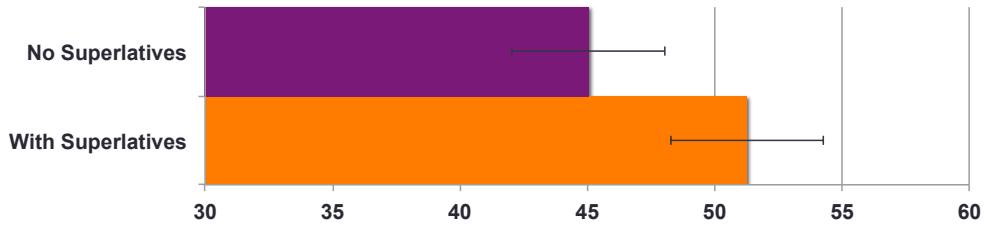

Figure 3.14: Guessing results using our original algorithm without superlatives vs. our original algorithm with the superlative attributes.

Figure 3.15: Examples adding superlatives as attributes to our original framework. For each image the left column shows the attributes selected by the algorithm when no superlative attributes where available, while the right column shows the ones in which superlative attributes were available. The green column shows two examples in which adding superlatives was helpful, while the red row shows two examples where adding superlatives produces lower quality referring expressions.

might involve answering a user's clarifying question, or providing feedback to a user who guessed incorrectly.

Finally, we believe our framework is an important component for any image description algorithm, though challenges remain dealing with integrate more general image descriptions (e.g., not just referring expressions).

CHAPTER 4

CONCLUSION

Generating natural language from visual data is becoming a more important task as we depart for the traditional paradigm of machines simply completing instructions and move to a more collaborative approach. There is a need in the computer vision field to start to investigate the new tools and algorithms necessary to facilitate this type of two way communication. In this thesis we focused on generating one of the most common expressions generated when using natural language: the referring expression. We have investigated factors which are important for generating these type of expressions for visual scenes.

More specifically we first investigated different measures of object saliency and how they effect the generation of referring expressions for visual scenes. We showed that when balancing object saliency and discriminability correctly we are able to generate more effective referring expressions than when simply focusing on discriminability.

We then proposed an efficient method for generating referring expressions under the condition of uncertain attribute classifiers. We first develop an algorithm which using the uncertainties can measure the confidence of a description in an efficient manner, and utilize this algorithm for generation. Through experiments we show that taking these uncertainties into consideration allows us to correctly measure the confidence of these descriptions.

Finally, we examine using additional tools available since we are using visual data. We show how incorporating absolute location in the form of location based attributes, relative location in the form of describing neighboring peo-

ple, and relative attributes in the form of superlative attributes can improve the quality of our referring expressions.

4.1 Future Work

The following are research directions I intend to pursue in my future career using the work presented in this thesis as a stepping stone.

4.1.1 General Image Description

Most previous works on general image description have focused on two main directions. First, there is the top down approach which attempts to utilize descriptions from previous images to new images. This approach usually yields very natural sounding sentences, but does not deal well with unexpected conditions (which are usually what we would want to describe in an image). Then there is the bottom up approach which, after detecting all the information in the image (scene category, objects, attributes) simply constructs sentences from all the information. Although this creates an image specific description, there is no way to prioritize what is more important to say.

It is obvious that a full general image description generator would need to include both directions. For example, one way to prioritize what is more important to describe in an image in the bottom up approach is by looking at what differentiates it from other similar images. If we have an image with green grass, perhaps it is not important to mention that the grass is green. However, if we observe an image with red grass this would probably be an important thing to

mention. In that sense we are merging top down knowledge (what is the usual color of grass) with the bottom up approach.

This idea of describing the unusual relates the general description generation problem to a referring expression generation problem. That is, in a sense every description is a referring expression trying to differentiate this image from all other images. Therefore it would make sense to use a similar mechanism to the one presented in this thesis for general image description by extending the context to be all other images. How to do so is not trivial since the set of distractors is extremely large, and new methods would need to be introduced.

4.1.2 Adding Additional Attributes

When generating referring expressions for objects we dealt solely with facial attributes. However, when describing people there are many other attributes which can be utilized which we have yet to examine. For example, clothing is an important attribute which can be much more visible than facial appearance attributes. Since the size of a clothing attribute can be much more prominent than a certain facial attribute it would make a lot of sense to utilize it. However, the main problem with using clothing is to generate a good clothing segmentation in order to determine which clothing belongs to which face. Especially in group photos this can be an extremely difficult task. We have begun to implement a robust clothing segmentation algorithm which will allow us to utilize this attribute in the future.

An additional attribute type which is possible to use, and which can prove very useful is pose based attribute. For example, if we can mention if someone

is standing or sitting, it might be a very visible attribute. However, this again would necessitate body segmentation for each person in addition to pose estimation. Although pose estimation has improved significantly in recent years it is still a very difficult problem in these crowded images. However, since our framework is general and can deal with imperfect attributes, we should be able to incorporate these attributers even before perfect classifiers are developed.

4.1.3 Expanding to Other Object Classes

Although when generating referring expressions for objects we dealt solely with people, the framework is general for any type of object. That is, just as we were able to generate referring expressions for people, we should be able to generate referring expressions for cars, buildings, cans, or any other object class without changing our algorithm. However, when attempting to transition to different objects there are a few issues which we expect to encounter.

First, there is the issue of object detection. Since face detection is robust we did not have to deal with uncertainties in our detector. However, since for most other objects this is not true, there will be a need to deal with object uncertainties as well. For example, if we want to describe a car, but there is a 50% chance that there is another car in the image, how do we deal with that? This would be an important question to answer, and a change to our basic framework would be necessary.

In addition there will be a question of what attributes to use, and which attributes are visible from different angles. When dealing with faces we had a large vocabulary of facial appearance attributes to use, which were always

visible since all the people were facing the camera. However, when dealing with new objects, we would need to examine first which attributes are relevant to that category. This can be done by examining previous descriptions of that object in text, to see what attributes people usually associate with this object class. Then, the framework would need to incorporate the fact that objects might be oriented in different angles and not all attributes will always be visible.

4.1.4 Applications

As described in previous sections there are many applications which could utilize an REG framework, some of which I hope to personally examine. For example, a visual assistant for the visually impaired using natural language. The idea is to use an egocentric vision system (such as the google glasses) whose camera observes the same visual scene as the user, to answer questions regarding the shared visual scene. This might include interactions such as:

Q: "Which can has beans in it?"

A: "It is the third can from the left on the top shelf"

Q: "Where are my keys?"

A: "On the the table next to the door"

Since the number of possible questions is endless, and it would be impossible to be able to answer all questions right away, we have developed a mobile application which would allow us to figure out which questions are important and feasible to answer. This mobile application allows visually impaired people

to take pictures of certain scenarios and then ask a verbal question. The image and question are then sent to AMT to allow users to answer it. By examining both the questions and answers we can find the questions we should focus on in the early stages.

Another possible application is integrating the REG framework to other robotic applications such as [36] or [38]. As mentioned in the introduction, as robot tasks become more complicated there is a growing need for flexibility and allowing the robots to ask questions in order to perform the task more effectively. Our framework would provide and important building block for robots to be able to achieve this type of communication.

APPENDIX A

WORKED OUT EXAMPLE

We provide a complete worked out example of our probability calculation as described in Sec. 3.2.3 & 3.2.4 of chapter 3.

A.1 Single Attribute

We start with the same example as given in chapter 3. Consider the case where a "smile detector" is applied to an image containing three faces, and without loss of generality, we describe face 0 as "the smiling face". What is the probability that a guesser, implementing the guesser model introduced in chapter 3, will be correct? To compute this, we must consider the fact that our smile detector is never certain, but instead, reports confidences of observing a smile on each face, as shown on the top of Fig. 3.3. The confidence associated with each score represents the probability that each face actually has a smile or not.

Continuing this example, the actual joint distribution of smiling faces in the image has $3^2 = 8$ total possibilities over the three faces (Column (a)), each with an associated probability (Column (e)). For each of these eight possible arrangements, the probability that the guessing strategy is correct can be computed (Column (f)). Naively, by applying total probability, the overall probability of guesser success is the sum of the probability that each of these eight smile cases occur, times the probability of guesser success in each case (summing over Column (g)).

For example, if all three faces are smiling (occurring with probability ($0.8 \times$

Number of smiling faces	0	1	2	3
Probability given face 0 is smiling	0	0.48	0.44	0.08
Probability of guessing correctly	-	1	0.5	0.333

Table A.1: The Poisson Binomial PMF of the number of faces with the correct attribute.

$0.4 \times 0.2 = 0.064$)), then the guesser has a 1-in-3 chance of guessing correctly. If face 0 is the only smiling face (which occurs with probability 0.384), then the guesser has a 100% chance of success. When none of the faces have smiles (occurring with 0.096 likelihood), the guesser chooses one face at random, and again has a 1-in-3 chance. By considering each of the 8 smile configurations, we compute the probability of a successful guess, $P_{\tilde{f}} = 0.613$. We can also compute the probability of a correct guess given the negative value "The person is not smiling" by simply taking the inverse of the top table and repeating the exact same steps ($P_{\tilde{f}} = 0.113$).

This method, computing the complete joint probability of attribute combinations, is correct, but it is also inefficient with complexity being exponential in the number of faces in the image and attributes in the description. Referring to our example from Fig. 3.3 for the attribute of smiling, we recognize that when our target face has the described attribute, the probability of a successful guess depends only on the number of the remaining faces that also have the described attribute, rather than the precise ordering. We can efficiently compute the distribution of the number of faces with the attribute using the Poisson Bernoulli Distribution, and the result is shown in Table A.1. The parameters of the distribution are simply the individual probabilities of smiling (Fig. 3.3 on top), except that we set the parameter for face 0 to 1 (since those are the cases we care about). Then to calculate the final probability we simply have to sum up

the product of the columns of Table A.1, multiply the sum by the probability that face 0 is smiling and add the case where no face has the correct attribute: $P_{\tilde{f}} = (0.48 \times 1 + 0.44 \times 0.5 + 0.08 \times 0.333) \times 0.8 + 0.2 \times 0.6 \times 0.8 \times 0.33 = 0.613$. This is exactly the same result as before, except the calculation is only polynomial in the number of faces (calculating the Poisson Binomial PMF has complexity $O(nlogn)$).

A.2 Multiple Attributes

For a multiple attribute description, we again use the efficient computation of the Poisson Bernoulli to compute $P_{\tilde{f}}$ in polynomial time. However, since in this situation we need to examine the cases in which only part of the attributes are correct for the target face we use the distribution in two steps. For example, Table A.2 shows the probabilities of two attributes. We wish to calculate the probability of a correct guess given the description: "the person is smiling and has glasses". First, we compute the probability of a face having n attributes correct for each $n \leq 2$. For each face this is a Poisson Binomial RV whose parameters are the corresponding column of A.2. Table A.3 shows the three PMF's, one for each face.

We now use Table A.3 to evaluate $P_{\tilde{f}}$ in a similar fashion to the single attribute. First, we can produce a table akin to Table A.1 for the meta-attribute of "satisfying 2 attributes". The bottom row is used as the probability of the attribute, and we repeat the same procedure as we have done for a single attribute. Table A.4 shows the probabilities for this meta-attribute. However, sometimes, no faces have all attributes (in our example this occurs with proba-

	Face 0	Face 1	Face2
Smiling	0.8	0.4	0.2
Glasses	0.7	0.7	0.2

Table A.2: Two attributes example

	Face 0	Face 1	Face 2
0 Attribute	0.06	0.18	0.64
1 Attribute	0.38	0.54	0.32
2 Attributes	0.56	0.28	0.04

Table A.3: Three Poisson Binomial PMF's, one for each face, over the number of attributes correct for that face.

bility $0.44 \times 0.72 \times 0.96$). To consider this case we produce a similar table to Table A.3 by removing its last row and renormalizing each column to sum up to 1. We use the bottom row as our probability and produce a similar table to Table A.4 but for the meta-attribute "satisfying one attribute". In the end all that is left is to multiply by the probability that all faces have one or less attributes.

Finally we sum up over all the different attribute numbers:

$$
\begin{aligned}
&(0.69 \times 1 + 0.30 \times 0.5 + 0.01 \times 0.33) \times 0.56 \\
&\text{When face 0 has two correct attributes} \\
&+(0.17 \times 1 + 0.58 \times 0.5 + 0.25 \times 0.33) \times 0.86 \times (0.44 \times 0.72 \times 0.96) \\
&\text{When face 0 has 1 correct attribute} \\
&+(0.06 \times 0.18 \times 0.64 \times 0.33) \\
&\text{When face 0 has no correct attribute} \\
&= 0.616
\end{aligned}
$$

This means that if for the case of Table A.2, if we say "the person is smiling and has glasses", a guesser should guess correctly with 61.6% chance.

Number of faces with two attributes	0 Faces	1 Face	2 Faces	3 Faces
Probability given face 0 has two attributes	0	0.69	0.30	0.01
Probability of guessing correctly	-	1	0.5	0.333

Table A.4: An example of a table akin to Table A.1 for the meta attribute "Satisfying two attributes".

BIBLIOGRAPHY

[1] A. Berg, T. Berg, H. Daume, J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, A. Sood, K. Stratos, et al. Understanding and predicting importance in images. In *CVPR*, 2012.

[2] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian. An o (n 2) algorithm for isotonic regression. *Large-Scale Nonlinear Optimization*, pages 25–33, 2006.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[4] R. Dale. Cooking up referring expressions. In *ACL*. Association for Computational Linguistics, 1989.

[5] R. Dale and E. Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 1995.

[6] R. Dale and E. Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[8] A. Farhadi, S. M. M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[9] M. Fernandez and S. Williams. Closed-form expression for the poisson-binomial probability density function. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(2):803 –817, 2010.

[10] A. Gallagher and T. Chen. Finding rows of people in group images. In *ICME*, 2009.

[11] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[12] P. Grice. Logic and conversation. *Syntax and Semantics*, 3:43–58, 1975.

[13] A. Gupta, Y. Verma, and C. Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012.

[14] B. G. H. Chen, A. Gallagher. What's in a name: First names as facial attributes. In *Proc. CVPR*, 2013.

[15] J. Harel. A saliency implementation in matlab. http://www.klab.caltech.edu/ harel/share/gbvs.php.

[16] H. Horacek. Generating referential descriptions under conditions of uncertainty. In *ENLG*, 2005.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1998.

[18] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012.

[19] E. Krahmer, S. Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72, 2003.

[20] E. Krahmer and K. van Deemter. Computational generation of referring expressions: A survey. In *Computational Linguistics*, 2011.

[21] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011.

[22] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. In *PAMI*, Oct 2011.

[23] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[24] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 2011.

[25] C. Mellish, D. Scott, L. Cahill, D. Paiva, R. Evans, and M. Reape. A reference architecture for natural language generation systems. *Natural Language Engineering*, 12(01):1–34, 2006.

[26] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. In *IJCV*, 2001.

[27] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[28] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[29] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012.

[30] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[31] A. Sadovnik, G. A., and T. Chen. It's not polite to point: Describing people with uncertain attributes. In *CVPR*, 2013.

[32] A. Sadovnik, G. A., and T. Chen. Not everybody's special: Using neighbors in referring expressions with uncertain attributes. In *The V&L Net Workshop on Language for Vision, CVPR*, 2013.

[33] A. Sadovnik, Y. Chiu, N. Snavely, S. Edelman, and T. Chen. Image description with a goal: Building efficient discriminating expressions for images. In *CVPR*, 2012.

[34] W. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012.

[35] M. Spain and P. Perona. Some objects are more equal than others: Measuring and predicting importance. In *ECCV*, 2008.

[36] S. Tellex, R. A. Knepper, A. Li, T. M. Howard, D. Rus, and N. Roy. Asking for help using inverse semantics. In *Robotics: Science and Systems*, 2014.

[37] J. Van De Weijer and C. Schmid. Applying color names to image description. In *ICIP*, 2007.

[38] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller. Learning semantic maps from natural language descriptions. In *Robotics: Science and Systems*, 2013.

[39] B. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 2010.

[40] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD*, 2002.

[41] Y. Zhang and T. Chen. Object color categorization in surveillance videos. In *ICIP*, 2011.